Ville Miekk-oja

# Static beacons based indoor positioning method for improving room-level accuracy

**School of Electrical Engineering**

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 28.4.2015

**Thesis supervisor:**

Prof. Jyri Hämäläinen

**Thesis advisor:**

D.Sc. (Tech.) Timo Vanhatupa

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Ville Miekk-oja

Title: Static beacons based indoor positioning method for improving room-level accuracy

Date: 28.4.2015 Language: English Number of pages:9+78

Department of Communications and Networking

Professorship: Networking technology Code: S-38

Supervisor: Prof. Jyri Hämäläinen

Advisor: D.Sc. (Tech.) Timo Vanhatupa

Demand for indoor positioning applications has been growing lately. Indoor positioning is used for example in hospitals for patient tracking, and in airports for finding correct gates. Requirements in indoor positioning have become more strict with demands for a higher accuracy.

This thesis presents a method for improving the room-level accuracy of a positioning system by using static beacons. As a static beacon, Bluetooth low energy modules will be used to test how much they can improve room-level accuracy on top of an existing positioning system. First, base technologies used in indoor positioning are reviewed. These include but not limited to: WLAN, Bluetooth and Ultra-wideband. Then, general indoor positioning are reviewed. After that, Ekahau positioning system will be introduced, as it is used as the base positioning system in the proposed method. The Ekahau positioning system applies WiFi networks and a fingerprinting method.

Next, tests and their results in the thesis are shown. In the first test, a received signal strength of a static emulated WiFi AP is measured indoors to study radio signal behaviour indoors and to investigate whether the proposed method could improve the room-level accuracy. Then in the second test, the proposed method is presented and tested with the Ekahau positioning system. The results showed that it is possible to improve the room-level accuracy by using static beacons.

Keywords: Indoor positioning, Bluetooth low energy, room-level accuracy, WLAN fingerprinting

AALTO-YLIOPISTO
SÄHKOTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ville Miekk-oja

Työn nimi: Sisätilapaikannusmenetelmä huonetarkkuuden parantamiseksi käyttäen staattisia radiomajakoita

Päivämäärä: 28.4.2015 Kieli: Englanti Sivumäärä:9+78

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Jyri Hämäläinen

Ohjaaja: TkT Timo Vanhatupa

Sisätilaipaikannusta hyödyntäville sovelluksille on ollut nouseva kysyntä. Tällä hetkellä sisätilapaikannusta käytetään muun muassa sairaaloissa muistiongelmista kärsivien potilaiden seuraamiseen, ja lentokentällä oikean lentoportin löytämiseen. Sisätilapaikannussovellusten vaatimukset ovat samalla kasvaneet. Tarkkuudessa, ja eritoten huonetarkkuudessa on ollut parantamisen varaa.

Tässä työssä esitetään menetelmä huonetarkkuuden parantamiseksi käyttämällä staattisia radiomajakoita. Radiomajakkana työn testeissä käytetään Bluetooth low energy -moduulia. Bluetooth low energy -moduulia testataan Ekahau paikannusjärjestelmän kanssa, nähdäksemme parantuuko huonetarkkuus moduulia käytettäessä.

Esiksi työssä esitetään teknologioita, joiden päälle monet sisätilapaikannusmenetelmät rakennetaan. Näitä ovat muun muassa WLAN, Bluetooth ja Ultra-wideband. Sen jälkeen käsitellään sisätilapaikannusmenetelmiä, jotka käyttävät edellä mainittuja teknologioita. Sitten esitellään Ekahaupaikannusjärjestelmä, jota käytetään tulevissa testeissä. Ekahaupaikannusjärjestelmä käyttää hyödykseen WLAN:ia, ja perustuu menetelmään nimeltä fingerprinting.

Lopuksi esitellään työssä tehdyt testit. Ensimmäisessä testissä mitattiin emuloidun WiFi -tukiaseman lähettämien signaalien voimakkuutta sisätiloissa. Testin tarkoituksena oli tutkia radiosignaalien käyttäytymistä sisätiloissa ja sitä, kuinka hyvät mahdollisuudet työssä esitetyllä menetelmällä voi olla parantaa huonetarkkuutta. Testin tulokset olivat positiivisia. Toisessa testissä testattiin huonetarkkuutta käyttäen Bluetooth low energy moduulia Ekahaun paikannusjärjestelmän kanssa. Testiskenaarion tuloksena oli, että huonetarkkuutta onnistuttiin parantamaan käyttämällä staattisia radiomajakoita.

Avainsanat: sisätilapaikannus, huonetarkkuus, Wibree, Bluetooth, WLAN

# Preface

I want to thank my Professor Jyri Hämäläinen for the continuous support I've got related to this thesis and work at Aalto in general.

I wish to express my sincere thanks to Ekahau for giving me this wonderful opportunity to do my thesis about indoor positioning. I want to thank my advisor Timo Vanhatupa, and my mentor Johannes Verwijnen, for all the guidance you gave me.

Last, I want to thank my family for supporting me throughout my studies.

Otaniemi, 28.4.2015

Ville V. V. Miekk-oja

# Contents

# Symbols and abbreviations

## Symbols

| | |
|---|---|
| $a_f$ | Attenuation factor for a floor |
| $a_w$ | Attenuation factor for a wall |
| $Bel(s_t)$ | Belief, that a system is at state $s_t$ |
| $c$ | Speed of light in a vacuum $\approx 3 \times 10^8$ [m/s] |
| $dist(l_i, l_e)$ | Distance function |
| $d_q$ | Minkowski distance |
| $D$ | Distance between an anchor node and a located device |
| $E(l_e\|x)$ | Positioning error estimation |
| $f$ | Frequency of a signal |
| $f_X(x)$ | Gaussian probability density function |
| $G_r$ | Located device's antenna gain |
| $G_t$ | Anchor node's antenna gain |
| $k_b$ | Slope of the side B |
| $k_c$ | Slope of the side C |
| $K(x; x_i)$ | Kernel function |
| $l$ | Location identifier |
| $l_e$ | Estimated location, for which an error-estimate will be calculated |
| $l_i$ | Location variable |
| $L$ | Path loss in Keenan model |
| $L_1$ | Path loss of one meter |
| $L_f$ | Floor penetration loss factor |
| $L_F$ | Free-space path loss |
| $L_{F(dB)}$ | Free-space path loss in decibels |
| $L_T$ | Path loss in ITU-R model |
| $M$ | RSS-matrix of a fingerprint in probabilistic approach |
| $n$ | Distance path loss exponent |
| $n_f$ | Number of floors |
| $n_w$ | Number of walls |
| $N_{EFF}$ | Effective number of particles |
| $p(l)$ | Prior probability |
| $p(l\|x)$ | Posterior distribution |
| $p(x)$ | Normalizing constant |
| $p(x\|l)$ | Likelihood function |
| $P_r$ | Measured received signal power at located device |
| $P_t$ | Anchor node's transmit power |
| $r$ | Radius of a circle and distance between anchor node and LD |
| $s_i$ | Online RSS measurement |
| $s_t^i$ | Discrete hypothesis of an object's location at time t |
| $S_i$ | RSS-vector of the detected signals from APs within a certain location, RSS of a fingerprint |
| $t$ | Time variable |

| | |
|---|---|
| $t_0$ | User's departure time from object A |
| $t_{diff}$ | Time difference |
| $t_{end}$ | User's time of arrival to object B |
| $t_{scan}$ | Scan time of WiFi APs' RSS |
| $w_t^i$ | Weight of a particle at time t |
| $x_{scan}$ | A location where a scan result of RSS from APs was received |
| $\eta$ | Normalizing constant |
| $\theta_{refract}$ | Departure angle of a refracted radio wave |
| $\lambda$ | Wavelength of a signal |
| $\mu$ | Mean value |
| $\sigma^2$ | Variance |

## Abbreviations

| | |
|---|---|
| AN | Anchor node |
| AOA | Angle Of Arrival |
| AP | Access Point |
| BAP tag | Battery-Assisted Passive tag |
| BT | Bluetooth |
| BLE | Bluetooth Low Energy |
| Blinds | Nodes that should be located |
| BTLE | Bluetooth Low Energy |
| ELP | Ekahau Location Protocol |
| ERC | Ekahau Real-time location Controller |
| ESS | Ekahau Site Survey |
| GPS | Global Positioning System |
| GSM | Global System for Mobile Communications |
| ID | Identification |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMU | Inertial Movement Unit |
| ISM | Industrial, Scientific and Medical |
| KLD | Kullback-Leibler Distance |
| KNN | K-nearest neighbors |
| LD | Located Device |
| LF | Low Frequency (30 - 300 kHz) |
| LOS | Line-of-sight |
| MEMS | Microelectromechanical systems |
| NIC | Network Interface Card |
| pdf | Probability density function |
| PDR | Pedestrian dead reckoning |
| RF | Radio Frequency |
| RFID | Radio Frequency Identification |
| RP-SMA | Reverse Polarity SubMiniature version A |
| RSS | Received Signal Strength |
| RSSI | Received Signal Strength Indicator |

| | |
|---|---|
| RTLS | Real-time location system |
| SHF | Super High Frequency (3-30 GHz) |
| SIG | The Bluetooth Special Interest Group |
| SISR | Sequential Importance Sample with Resampling |
| SMG | Spinning mass gyroscope |
| SNR | Signal to Noise Ratio |
| TDOA | Time Difference Of Arrival |
| TOA | Time Of Arrival |
| USB | Universal Serial Bus |
| UTD | Uniform theory of diffraction |
| UWB | Ultra-wideband |
| WGAN | Wireless Global Area Networks |
| WLAN | Wireless Local Area Networks |
| WWAN | Wireless Wide Area Networks |
| XML | EXtensive markup language |

# 1 Introduction

Indoor positioning has been a widely researched subject for over decades. Despite the fact that indoor positioning has not yet been witnessed as such a mainstream technology as outdoor positioning with Global Positioning System (GPS), the industry of indoor positioning applications has been growing. Currently there are many indoor positioning applications such as patient tracking in hospitals, asset tracking and navigation.

Outdoor positioning is mainly done with GPS that uses signals from satellites. Indoors, the walls of buildings greatly diminish the GPS satellite signals which causes inaccurate positioning. Therefore, other technologies have been used and studied for indoor positioning. These include Wireless Local Area Networks (WLAN), Bluetooth (BT), Ultra-wideband (UWB), Global System for Mobile Communications (GSM) and even the use of Earth's magnetic field.

This thesis will focus mainly on positioning methods that can be used with WLAN and Bluetooth technologies, but general indoor positioning methods that can be applied on other technologies will be covered as well. One of the most common indoor positioning method for WLAN is received signal strength (RSS) -based fingerprinting. RSS-based fingerprinting is based on measuring radio signals' strength from multiple Access Points (AP) and determining a position by comparing the RSS to a fingerprint model of an area.

Fingerprinting-based positioning has two phases. A radio map of RSSs from APs is first created for the location. This can be done for example, by making measurements in predetermined spots and storing the location information and corresponding RSS values. In the second phase, RSS values are measured from the APs and then compared to the fingerprints that were stored earlier. Based on this comparison, the position with the closest match is chosen as the correct position. There are different methods on how to calculate the closest match, and they will be reviewed later.

The fingerprinting method has some drawbacks unfortunately. One of them is that the fingerprinting-based methods are not reliable in places that share a similar fingerprint. A similar fingerprint in two places means that the measured RSSs from APs are close to equal. This can happen especially with two neighbouring rooms close to each other. One solution is to add more traditional radio beacons (Wifi or Bluetooth) to the area to increase fingerprint uniqueness, but that can become expensive in the long run and the area would have to be recalibrated.

## 1.1 Thesis scope and objectives

The accuracy of a WiFi fingerprinting based positioning system can decrease, if two or more places share a similar WiFi fingerprint. This thesis presents a method for improving a room-level accuracy in a WiFi fingerprinting based positioning by using low-cost static radio beacons. When low-cost static radio beacons are added to an existing positioning system, the fingerprints of a radio map become more unique. More unique fingerprints can increase the room-level accuracy. In the context of

this thesis, the room-level accuracy is quantified as the accuracy that a positioning system can locate a device to a correct room. In the context of the room-level accuracy, it doesn't matter where inside the room the system locates the device, as long as it can locate devices correctly on a room level. Static radio beacons can be also used in a way that adding them to the positioning system doesn't necessarily require a recalibration of the area of interest.

The main question is, can low-cost static radio beacons increase the room-level accuracy in a Wifi fingerprinting-based system? In this thesis, the static beacons based indoor positioning method will be tested on top of real a indoor positioning system. As the low cost static radio beacons we will use Bluetooth Low Energy (BTLE) modules and WiFi Universal Serial Bus (USB) injectors with an attenuator that simulates a low-energy wifi beacon.

## 1.2 Thesis structure

The remaining thesis work is divided into four sections. In section 2 we make a literature research about the base technologies used in indoor positioning. Different base technologies provide possibilities for different indoor positioning methods. Some of the positioning approaches can be used with many different technologies, while some of them are very technology specific. As for the base technologies, Wifi, Bluetooth, Radio Frequency Identification (RFID), UWB and Inertial Movement Unit (IMU) are reviewed. Then, in section 3, the positioning approaches that use some of these technologies for indoor positioning are reviewed. These include trilateration, angulation, Time Difference Of Arrival (TDOA), Time Of Arrival (TOA), Wifi fingerprinting and filtering. Wifi fingerprinting is given more emphasis, since it is the base method used in the tests of this thesis.

In section 4, the most essential characteristics of a radio signal propagation indoors are described. Indoor propagation is in a very essential role in indoor positioning, and in the new method we propose. Different path-loss models, multipath propagation, shadowing and small-scale fading are covered.

In section 5, Ekahau positioning system is presented. The Ekahau positioning system is an implementation of a WiFi fingerprinting method. The proposed method in the thesis is tested on top of the Ekahau positioning system.

Test scenarios and their results are reviewed in section 6. Two different kind of test were made: RSS-measurement tests and room-level accuracy test. In the RSS-measurement test, the RSSs of an emulated WiFi AP are measured indoors in order to study indoor propagation of radio signals and to investigate whether the proposed method of static beacons could improve the room-level accuracy in theory. In the second test, the room-level accuracy of Ekahau positioning system is tested with and without the aid of a BTLE beacon. First, the integration between BTLE and ekahau positioning system will be shown. Then, the methodology, results and result analysis of the room-level accuracy test will be reviewed.

In the last section (7), summary of the thesis is presented with future work suggestions that could be made to further improve the proposed method and tests.

# 2 Technologies used in indoor positioning

In this chapter we will shortly review few of the base technologies that are currently used and studied for indoor positioning. These include WLAN, Bluetooth, RFID, UWB and the use of IMU. Many of these technologies were not originally designed to be used in indoor positioning. For example, WLAN was designed for wirelessly connecting computers. Majority of indoor positioning methods are based on the use of radio signals. Therefore, many of the approaches require a device capable of receiving radio signals, and radio beacons that transmit radio signals. Before we review the positioning approaches that can be used with these technologies, let's cover the base technologies.

## 2.1 Wireless Local Area Networks

WLAN is a popular technology for wireless connections between devices and connecting clients to infrastructure networks [1]. A wireless connection is established and maintained by creating electromagnetic waves, which propagates through air. WLAN is currently used in four different radio frequency zones: 2.4 GHz, 3.6 GHz, 4.9 GHz and 5 GHz. The most common frequency zone is 2.4 GHz. The majority of modern WLANs are adhering to Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards, commonly labelled as Wifi. Mobility is a significant advantage of WLANs. Users can be mobile while connected to each other and to a fixed network. The architecture and technical overview of WLAN can be found in [2]. Other wireless technologies that can provide the same aforementioned functions as WLAN are for example, Wireless Global Area Networks (WGAN) and Wireless Wide Area Networks (WWAN). In *Lindroos*[2], a comparison is made between WLAN, WGAN and WWAN. WLAN is seen as the best choice for the residential buildings for the following reasons:

- WLAN-coverage is good for many indoor environments, especially for residential buildings. For security reasons the optimal signal strength should be strong inside the building, and weak outside. WLAN-coverage is usually optimal for this.
- Because users often want to connect to the Internet, the throughput of the technology has to be high enough. This can be achieved with WLAN.
- The cost of WLAN is very low.

Because WLAN has become so popular nowadays, that it already exists in many indoor infrastuctures, it is naturally very tempting to use its radio signals for indoor positioning. This way an indoor positioning system can be implemented without the need for installing any additional equipment. Though in practise, the density of WLAN APs (Access Points) can vary a lot in the area. If the density of APs is low, then there is a need to install additional APs. Generally speaking, the higher the amount of APs, the better the conditions for positioning methods to work. Locations of the APs have an impact on the positioning system as well. Also the APs channels

should be selected in such a way that they don't interfere with each other. On 2.4 GHz WLAN, one very common choice is to use channels 1, 6, 11 and 14 together, because they don't overlap with each other.

## 2.2 Bluetooth

Bluetooth is another wireless technology for connecting devices and exchanging data over short distances. It is standardized in IEEE standard number 802.15.1, but that standard is no longer maintained. Currently Bluetooth is managed by The Bluetooth Special Interest Group (SIG), which includes more than 19000 member companies in the areas of networking, computing, telecommunication, and consumer electronics.[3] Idea behind Bluetooth is to enable low-power radio communications to link computers, phones and other kind of network devices over short range. Motivation for bluetooth was to replace the cable(s) that were used in connecting fixed or portable electronic devices.

Next, lets take a look at some of the technical specs of Bluetooth. Bluetooth uses short-wavelength microwave transmissions in the Industrial, Scientific and Medical (ISM) band from 2400Mhz to 2480Mhz. Frequency hop tranceiver is used in Bluetooth to diminish fading and interference. The Bluetooth protocol uses a combination of packet and circuit switching. Good overview of the Bluetooth architecture and features can be found in [4].

Differences between WLAN and Bluetooth are:

- Bluetooth signal has generally a lower distance range.
- Bluetooth costs less to implement. Also the power consumption of Bluetooth is generally lower.
- Wifi has a bitrate many times higher than BT.

### 2.2.1 Bluetooth Low Energy

The BTLE was officially introduced in 2011 in Bluetooth Specification v4.0. Bluetooth Low energy has much lower power consumption than in classic Bluetooth, has new features and unique characteristics. This enables new types of applications that were not practical with Classic Bluetooth. BTLE was originally devised by Nokia under the name of Wibree. With a low power consumption, BTLE can be used in small devices with a tiny coin cell battery, that can last for 1-10 years, depending on usage. BTLE is aimed at applications in security, fitness, healthcare and home entertainment industries. [4]

### 2.2.2 Bluetooth and positioning

Bluetooth contains useful characteristics which can be used in indoor positioning. Because of the low distance range of Bluetooth signals, Bluetooth can be utilized in cell-based positioning [5]. Bluetooth cell-based positioning contains Bluetooth

transmitters covering the localization area such that each transmitter and its signal range forms a cell area, as demonstrated by Figure 1.

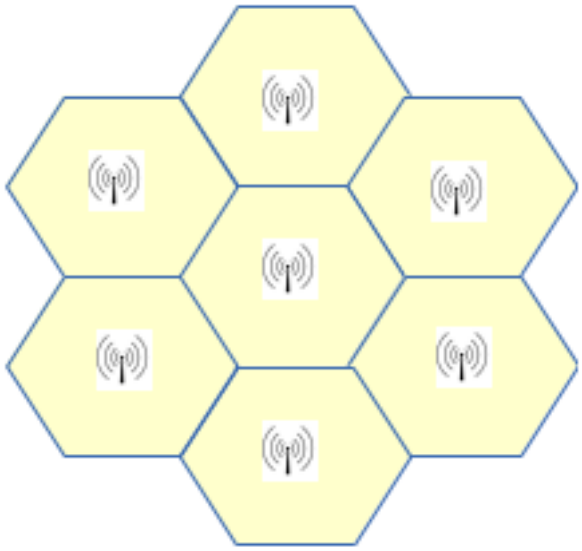

Figure 1: Cell based positioning structure

This kind of cell-based infrastructure enables tracking of a device capable of receiving Bluetooth signal. The device can be located in the cell area of which Bluetooth signal it is detecting the strongest. Of course this method requires a large amount of BT transmitters, but because BT transmitters cost little and lifetime is high, it is a financially scalable solution. Its viability increases especially with BTLE, that has even lower power consumption. One of the goals of this thesis work is to utilize and study BTLE beacons to gain more room-level accuracy for indoor positioning. The communication range of Bluetooth is generally less than WLAN, but based on my personal observations during measurements, in many cases the signal can be detected equally far away. Detecting weak BT signal doesn't allow good connection and data transfer between devices, but this doesn't mean the signal could not still be used in positioning. This is why many other common positioning methods also apply to Bluetooth. These positioning methods will be discussed further in the next section.

## 2.3 Radio Frequency IDentification

Radio Frequency IDentification (RFID) is a method mainly for tracking and identifying tags attached to different objects. Commercially, RFID is often used as a replacement for bar codes [6]. RFID is a wireless and a contact-free method, that uses radio-frequency electromagnetic fields for transferring data between RFID tags. RFID also has an option to store data to a Radio Frequency (RF) compatible integrated circuit [7]. RFID systems operates on many frequency bands from Low Frequency (LF) ( 30 - 300 kHz ) to Super High Frequency (SHF) (3-30 GHz).

An RFID system consist of three components: a tag, a reader and antennas in both. RFID tags can be either passive, active or passive with battery assistance. A passive tag has no integrated power source. It only activates when it receives a signal from an RFID reader that will power it up to send back a response signal. Active tags have their own battery power source, that can be used to periodically transmit ID signal. A battery-assisted passive (BAP) tag works similarly to the passive tag, except that it doesn't necessarily need an external power signal from the reader, since it has its own battery source. A BAP-tag doesn't actively transmit an ID signal, but it is activated when it detects an RFID reader. Because passive tags have no internal power source, the range of their signal is usually lower than that of an active tag. RFID tags can be either read-only, or read/write enabled, where the user of the system can write data to them. [8]

An RFID reader is used to read tags and get information from objects attached to tags. An RFID reader produces energy to passive tags by using its electromagnetic field. Passive tags can then use this energy to send information to the reader. Readers can be also either passive or active. Passive readers receive signals from tags, but don't transmit anything by themself. A system with passive readers requires active tags that transmit their ID signal periodically. Active readers transmit interrogator signals to tags, and receive information from them. [8]

The RFID tags and readers can be made very small and cheap and their deployment and usage in different devices and places is straigthforward and don't necessarily require much maintenance. Currently RFID-based systems are widely used in libraries and big malls for disbursing and monitoring. Also, tags can be implanted for tracking of animals, for example. RFID tags can be made very small, as seen in Figure 2.

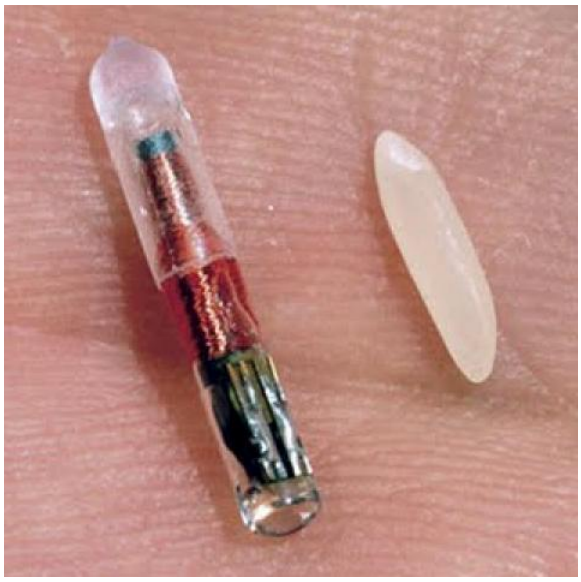

Figure 2: RFID tag size comparison with rise grain [9]

RFID-based systems have been used in indoor positioning as well. Contrariwise to the normal RFID system use case, in the indoor positioning case the actual

reader is the device usually being tracked or located. A very common scenario is to have static RFID tags scattered over the area. The RFID tags contains a location information that tells where the tag is located. Then a moving object will carry an RFID reader that reads signals transmitted by tags. If the tags are passive, then the reader also sends power signal for them. Based on the received signals from the tags, the readers position can be estimated. Many different kind of methods and algorithms for RFID have been invented. A common one is cell-based positioning by using RSSIs (Received Signal Strength Indicator) of the tags. The reader can be located to an area of which the RFID tag's RSSI value reader detects the strongest. [10]

An extensive state-of-the-art review [11] of RFID-based positioning methods includes the use of RSSI information and combining RFID-based positioning with Pedestrian Dead Reckoning (PDR). The authors' own proposed approach in [11] is localization based on RSSI measurements combined with RFID tag detection count probability. They estimate distances between all the tags and the reader from the RSS based on radio signal attenuation model. These distances are then used by a particle filter to get more accuracy. Traditional positioning methods like trilateration and fingerprinting can also be used in RFID technology, as presented in [12].

## 2.4 Ultra-wideband

Ultra-wideband, as the name indicates, uses a wide frequency band for communication with radio waves. UWB can be used for short distances with low energy and high bandwidth. UWB is used in non-cooperative radar imaging, tracking, localization, and sensor data collection.[13] In military, UWB radars have been used for multiple decades [14, 15].

Bandwidth of UWB can be wider than 500Mhz in absolute frequency. Remarkable difference with traditional radio transmissions and UWB is, that UWB modulates the information by generating radio energy at certain time intervals, which is called time modulation. Wide band gives handfull of benefits in radar applications, communications and positioning. One benefit is that UWB contains multiple different frequency components, which increases the probability of UWB signal to penetrate through walls. This is because the frequency that passes through obstacle is dependent on the obstacle's material. Different material passes different frequencys and prevents others. Small-scale fading effect, known for radio communications, is also decreased when using large bandwidth. [13]

The signal penetration property of UWB provides quality and opportunities in indoor positioning. It enables positioning methods that can not be used very well with other technologies in indoors. These include TDOA and TOA -trilaterations and Angle Of Arrival (AOA). These techniques will be explained further in this thesis. Multipath distortion of radio signals caused by reflections from walls and obstacles is a common hindrance for TOA and AOA -based positioning techniques in traditional RF positioning systems. TOA and TDOA methods require the original unreflected signal in order to work properly. Because UWB pulses have a very short duration, it is possible to filter out the reflected signals from the original signals

[16]. The advantages of UWB technology in indoor positioning are: less interference, good material penetration of signals, no multipath distortion, doesn't require a line-of-sight, UWB sensors are cheap and they have a large coverage range [16]. The accuracy scale of UWB positioning system is generally in centimeters.

## 2.5 Inertial Measurement Unit

Inertial measurement unit is a device that consist a combination of gyroscopes, accelerometers, magnetometers and a barometer. For device to be labelled as IMU, it doesn't require to have all of the aforementioned gadgets. IMU unit measures velocity, orientation, air pressure, and gravitational forces of the object it has been attached. These information can be utilized for example in navigation and positioning. Originally IMU was used a lot in aircraft navigation, and it is still used as a backup system if GPS fails. In indoor positioning, IMU is utilized in a positioning method called pedestrian dead reckoning. Next, lets review the components of the IMU unit. [17]

### 2.5.1 Gyroscope

Gyroscope (Fig. 3) is a device capable of sensing the changes in orientation and angular velocity. There are different ways how to implement a gyroscope. Figure 3 represents a classic mechanical spinning mass gyroscope. A spinning mass gyroscope (SMG) has a spinning disk that is mounted into a loosely rotating joints. When the disk is set to high reeling motion, it remains its orientation almost the same even if the gyroscope is turned around differently. Reason for this is in the angular momentum the disk holds. Disk with angular momentum tries to resist any movement in its orientation. Because the disk remains its orientation even if gyroscope's orientation is moved, we can then easily sense the changes on the objects orientation the gyroscope is being attached to. For positioning purposes, this will give us information about the changes of orientation. The more information we have about the movement of object being positioned, the better we can estimate its position. [18]

Two other common gyroscopes are vibrating mass gyroscope and optical gyroscope. Vibrating mass gyroscopes are popular Microelectromechanical systems (MEMS) gyroscopes, that can be found for example inside smart phones. Vibrating mass gyroscope works by sensing angular velocity from two different modes of vibrations. First vibration mode is called drive-mode, and in drive-mode the mass is set to vibrate in a linear way using electronic current. The second mode is called sense-mode, and it is caused by the Coriolis effect caused by drive-mode vibration and input of external angular velocity. From the displacement of gyro's inner structures, the amount of external angular velocity can be sensed. [20]

### 2.5.2 Accelerometer

In IMU based indoor positioning, accelerometers are in very crucial roles. Accelerometers used inside devices like smartphones are micromachined accelerometers. They

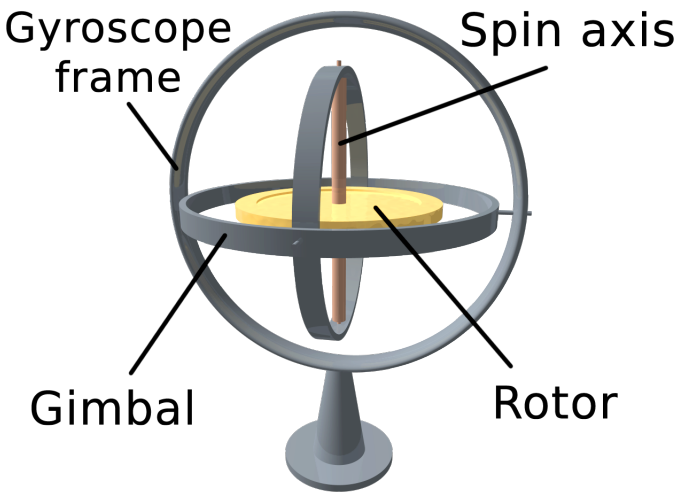


Figure 3: Gyroscope [19]

consist a supported mass, outlying frame, and springs between the frame and the mass. Accelerometer can be modeled as in Fig. 4. When no forces are influencing the system, the springs have an equal tension and the proof mass's position will be in the center of the frame. If the frame is under acceleration, the mass will displace from the center which causes electrical change. Acceleration of the frame can be determined from this electrical change. Usually one IMU contains 3-axis accelerometer, so acceleration can be sensed in 3 dimensions. [21] Regarding of indoor positioning, acceleration can be used to estimate travelled distance by counting foot steps.

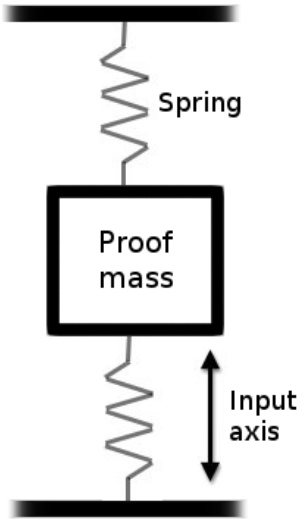


Figure 4: Modelled accelerometer

### 2.5.3 Magnetometer and barometer

Magnetometer is a device for measuring either the strength of the magnetic field, direction of the field, or both. In general, magnetometers can be split into two categories: vector and scalar magnetometers. Vector magnetometers measure the strength and direction of the magnetic field, whereas scalar magnetometers only measure the strength. Carl Friedrich Gauss invented the first magnetometer in 1833. [22]

Sensing the direction and strength of magnetic field can both be used in indoor and outdoor positioning. Knowing the direction of magnetic field, one can use the magnetometer like a compass to get heading information. If we consider small device with magnetometer carried by human, the device should be mounted into one position in order to get proper heading information from magnetometer. The strength of magnetic field can be utilized in fingerprinting based positioning. [23]

Barometer is a device used for measuring atmospheric pressure. It is commonly exert in measuring air pressure to forecast changes in the weather. In indoor positioning, a barometer can be used for detecting correct floor level. Quick changes in air pressure can indicate vertical movement. [24]

# 3 Indoor positioning methods

In the third chapter of this thesis the most common indoor positioning approaches are covered. Many of these methods are usable in outdoor positioning as well. Weaknesses and strengths of the methods will be analysed. The methods include: Trilateration, triangulation, TDOA and TOA, RSSI-based ranging, fingerprinting and filtering. The methods are not exclusive to each other. One could use many of them simultaneusly to increase the performance of a positioning system. The use of different filtering techniques has become popular in indoor positioning. Based on history estimates, a positioning system can filter out false positioning candidates, and therefore increase the positioning accuracy. In section 3.6, we will review a simple filter, which is based on error estimate, and a particle filter.

Many of the positioning approaches consist of similar system components. An anchor node/reference point is a one typical component of an positioning system. These reference points provide crucial information, that is then used in positioning. It is common, that the location of an anchor node is known beforehand. Usually, the devices wanted to be located and anchor nodes exchanges information with each other. The content of this information is then applied in localization. In the context of this thesis, the nodes or devices wanted to be located are called ***blinds***.

## 3.1 Trilateration

Trilateration is a positioning approach, that can be used to locate an object by using the geometry of spheres. Requirement for the trilateration approach is, that the distances between a blind node and at least 3 reference points has to be known. There are different means how the distances can be estimated. If the blind node and the reference points are RF compatible devices, one can calculate the distance based on signal strength attenuation. Different propagation models exists that can be used for that. Another way to obtain the distance is by calculating a radio signal's flight time between an anchor node and the blind. This method is called Time of Arrival (TOA). Measuring the distance based on TOA and received signal strength (RSS) attenuation will be reviewed later.[25]

Once the distances between the anchor nodes and the blind are known, lateration process can be applied. In lateration, the blind is located by using the geometry of circles. Let's consider a situation where we have one anchor node and a blind, and we know the distance (d) between them, but not the heading of anchor node. What we do know, is that the blind is located somewhere on a circumference of a circle that has a radius of d. To get the exact position, we need to know the distances between of at least 3 different anchor nodes and the blind. If we draw the 3 circles that forms from the distances between the anchor nodes and the blind as in Fig. 5, we can see that they intersect in one spot, which is the correct position. [25]

The intersection point can be found by using mathematics. We have three circles and we just need to calculate where their x and y coordinates are equal. First, we have the three circles as in Equations 1, 2, 3.

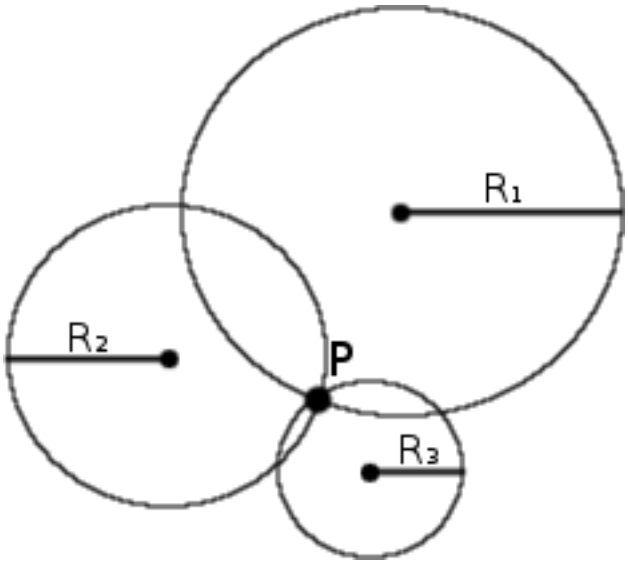


Figure 5: Trilateration

$$(x - a_1)^2 + (y - b_1)^2 = r_1^2 \quad (1)$$
$$(x - a_2)^2 + (y - b_2)^2 = r_2^2 \quad (2)$$
$$(x - a_3)^2 + (y - b_3)^2 = r_3^2 \quad (3)$$

Two circles will intersect at two points. The first two circles will have two pair of x and y coordinates as their intersection points. These can be solved for example by solving y from the Equation 1, and inserting it to Equation 2. Then x can be solved from Equation 2. X will have two values, that can be used to calculate the two y coordinates. Once we know where two circles intersect, we can look where the third one will intersect. The third one will intersect two times with both of the circles making it total of 4 intersections. Two of these intersection coordinates will be the same coordinates where circles 1 and 2 crossed. This point is the point where all circles intersect, thus we have found our goal.

In practise, the three circles never really intersect exactly at one spot. This is because we never get absolutely right distances between the anchor nodes and a blind. Especially in indoors, the environment is very challenging for estimating distances between anchor nodes and blind nodes. Challenges for trilateration are (1) variations of wireless communications; (2) incorrect measurement of range; (3) dynamic environment with disturbance [25]. Different technologies and different methods can affect a lot for the performance of estimating distances as we will see in the forthcoming sections. Indoors, the trilateration method struggles in situations where there are no-line-of-sight between a blind node and anchor nodes.

## 3.2 Angulation

Angulation positioning method is based on arrival angles of signals. [26] These signals can be for example electromagnetic signals or sound waves. The method is often also referred as Angle of Arrival (AoA). Like trilateration, angulation based method can be used in a system that has anchor nodes (AN) and blind nodes. In Angulation, the anchor nodes doesn't necessarily need to send any signals to the blind. Common scenario is that blind node is the one sending signals to anchor nodes. Requirements for angulation positioning method to work in two dimensions are following [27]:

- At least two anchor node positions have to be known.
- Anchor nodes need to be able to measure the arrival angles of the signals originated from the blind nodes.
- Direct signals are required; reflected can cause errors.

Once we know the arrival angles of the signals, we can draw those signal lines and the point where the lines cross is the estimated location. Let's go through how by using mathematics we can get the position coordinates of a blind node with two anchor nodes as in Fig. 6. In Fig. 6 the anchor nodes are marked as AN and the blind node is marked as LD (Located Device). Goal is to get the coordinates of the blind node. Since we beforehand know the locations of ANs, we know the distance between them, marked as C. We also know the slope and function of side C. Probably easiest way to get coordinates of the blind node is to calculate the functions of side A and B, and then see where they intersect. Lets calculate the function of side B. Slope of C is:

$$k_c = tan\theta$$

Because we know the angle between sides C and B, and the slope of C, we can get the slope of B:

$$k_B = tan(\theta + (\pi - \alpha))$$

The function of side B has a form of:

$$y = k_B x + R$$

We already know the slope of B, so we just need to figure out the constant R. That can be solved by using the fact that function B goes through point AN. Lets mark the point AN as: $(x_0, y_0)$. We can now insert that point to the function of side B and solve the constant R:

$$R = y_0 - k_B x_0$$

So the function of side B is:

$$y = x * tan(\theta + (\pi - \alpha)) + y_0 - k_B x_0$$

Next we can do the same thing to calculate the function of side A, and then just calculate where they intersect. That operation is mathematically trivial and left therefore outside of this thesis.

The pros of angulation method is, that no calibration of the area is needed beforehand. Cons and challenges are:

- Currently can not utilize existing wlan architechtures. One has to install separate special hardware.
- Multipath effect in indoors affects to the angle of arrival; signal can arrive from multiple directions.

Before in this thesis we mentioned that Ultra-wideband technology uses very fast pulses, which enables the detection of different multipath components. In AoA this can be utilized to find the direct path component or at least migitate the errors of multipath reflections. Overview of migitation methods and multipath reflection problems can be found in [28].

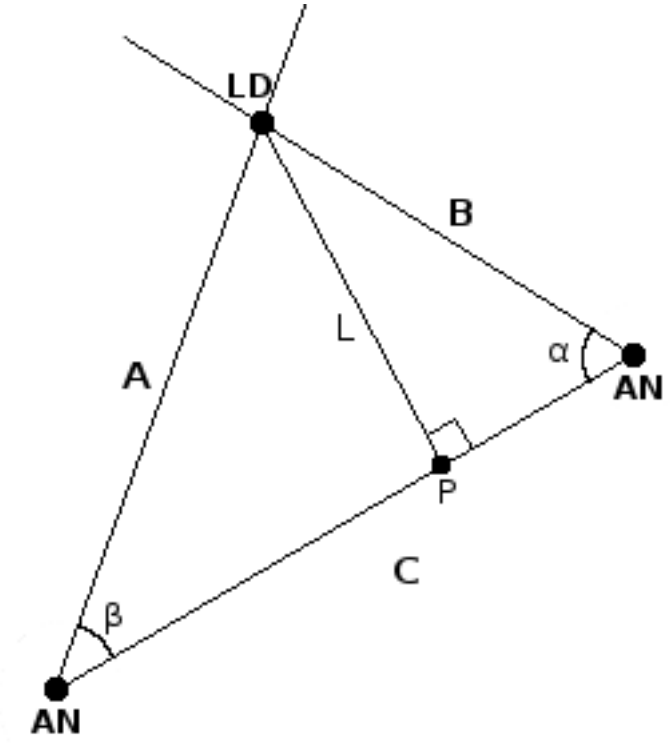


Figure 6: Angulation

## 3.3 TOA and TDOA

Time Of Arrival (TOA) and Time Difference Of Arrival (TDOA) are methods that try to estimate a location of a blind node based on the travel time of a signal. These methods can be used in a system that has multiple anchor nodes and blind nodes. In both methods, the blind nodes has to send signal to anchor nodes. Lets review TOA first.

In TOA, the flight time of signals traversing between anchor nodes and blinds are calculated to get distances. TOA method can only give distances between blinds and anchor nodes. The estimation of the location can then be obtained by using for example *trilateration* method, discussed earlier. One way to calculate the time of flight is following: a blind node sends a radio signal to the anchor nodes. The radio signal consist of a timestamp message of the signal's time of departure. When an anchor nodes receives the message, it writes down the arrival time and calculates the time difference by subtracting the departure time from the arrival time. The TOA approach requires precice clock synchronization between the blind nodes and the anchor nodes. Clock synchronization is one of the challenges involved in the TOA. [27]

Once the time difference has been calculated, the actual distance between the anchor node and the blind can be calculated simply by formula:

$$D = c * \triangle t$$

Where c is the speed of light in medium, and $\triangle t$ is the time difference.

In TDOA method, a blind node sends signals which anchor nodes can detect. The location of the blind is estimated by using the time difference of the arrival signal between several anchor nodes. Lets have an example in two-dimensions, where we have two anchor nodes and a blind node, that is sending signals. When the anchor nodes receive the signal originated from the blind node, they will save the arrival time into memory. Then the difference between those arrival times is calculated. We know that the blind has to be located on places that share the same time difference. Because now time is linear to distance, the possible locations of the blind are those where the difference of distance remains the same. When these possible locations are calculated mathematically, we actually find out that they are all located on a hyperbola.[29]

In order to get the exact position, we need to add a third anchor node to the system. Then we can calculate a different hyperbola of possible locations. The exact position of the blind node can then be found in the intersection point of the two hyperbolas.

The benefit of TDOA compared to TOA is that in TDOA, only the anchor nodes' clocks needs to be synchronized between each other [27]. It doesn't matter whether the blind nodes are in sync or not, since it doesn't affect to the time difference of arrival. Indoors, Both TOA and TDOA methods suffer from multipath reflections which causes the signal to arrive later than expected. But with UWB technology, different reflection components can be detected and filtered, making TOA and TDOA noteworthy method for also in indoor positioning. [30]

## 3.4 RSSI-based ranging

In RSSI-based ranging the distance between a blind and an anchor node is estimated from the signal path loss by using radio signal propagation models [31]. The path loss between a pair of antennas is defined as the ratio of the transmitted power to the received power [32]. Lets assume we have an anchor node, that is transmitting radio signals captured by a blind node. When the signal output power of the anchor node is known, the attenuation of the signal can be calculated by measuring the received power at the blind node and then subtracting that from the signal output power. The distance is then calculated based on a chosen propagation model. Once we know the distances from the blind node to three or more anchor nodes, a location of the blind can be calculated by using the *trilateration* method discussed earlier.

There are several propagation models invented for radio signals. In a free space, the path loss is [32]:

$$L_F = \frac{P_t G_t G_r}{P_r} = \left(\frac{4\pi r}{\lambda}\right)^2 = \left(\frac{4\pi r f}{c}\right)^2$$

Where:

- $L_F$ is the free space loss.
- $P_t$ is anchor node's transmit power.
- $G_t$ is anchor node's antenna gain.
- $P_r$ is a signal's received power at a blind node.
- $G_r$ is blind node's antenna gain.
- $r$ is the distance between anchor node and a blind node.
- $\lambda$ is the wavelength of the signal.
- $f$ is the frequency of the signal.
- $c$ is the speed of light.

The free space loss model expressed as decibels is:

$$L_{F(dB)} = 32.4 + 20logR + 20logf_{MHz}$$

Where $L_{F(dB)}$ is attenuation in decibels, R is distance in kilometers, and $f_{MHz}$ is frequency in megahertz.

However, in most of the cases these forementioned equations don't provide accurate estimation of the distance under real-life conditions [33]. In indoor environment, a radio signal can experience many physical phenomens, which makes it more hard to estimate a distance based on RSS. Indoors can contain walls, objects and moving people for example. These all have an effect to the signal strength. The signal

can fade or reflect, which causes multipath issues. Therefore, more advanced signal propagation models have been invented for indoors. These include, for example, Log-Distance path loss model, ITU path loss model and Keenan and Montly path loss model. We will come back to these path loss models in section 5, where we discuss about indoor propagation and phenomenons. Despite the more advanced indoor propagation models, the accuracy of the RSS-based ranging is generally seen as moderate due to its physical constraints.[27]

## 3.5 Fingerprinting

Fingerprinting is an indoor positioning technique, where a map of fingerprints is first created for the location [34]. Then online measurements are compared with the fingerprints, and the location of a blind node is determined based on the comparison. What are these fingerprints then? Generally speaking, fingerpints are location-dependent characteristics mapped to the location [31, 16]. In case of WLANs, possible fingerprints can be for example, the RSSI of the APs. Other example attributes that have used as fingerprints are:

- Radio signal's signal to noise ratio (SNR) [35].
- The Earth's magnetic field [36].
- Air pressure [37, 38].
- Sound waves [39].

As a fingerprint, it is possible to use all the available location-dependent characteristics together. In the context of the remaining thesis, the term fingerprinting means the Wifi RSSI-based fingerprinting. That is the base method used in our test system, and therefore we will focus more on it in this thesis. Fingerprinting has two phases, an offline phase and an online phase. In the offline phase, a radio map of RSSI fingerprints is created for the location. In the online phase, the current measurements are compared to the radio map and closest match is selected as the positioning estimate. First, lets review offline phase. [34]

### 3.5.1 Offline phase

In the offline phase, a radio map of fingerprints is created. Lets define a fingerprint in the Wifi RSS-based fingerprinting. Content of a fingerprint is dependent on the approach of the chosen comparison method. Two very common approaches are a deterministic and a probabilistic approach, which we will review further in the thesis. In the deterministic approach the fingerprint can be defined as $(l, S_i)$, where $l$ denotes the location identifier, and $S_i$ is an RSS-vector consisting all the RSSIs from detected APs in the location [40]. In other words, $S_i = (S_1, S_2, S_3, ..., S_n)$, where $n$ is the amount of detected APs. In deterministic approach it is common to use the average value of RSS from each AP. This means that RSS of each detected AP is measured multiple times at each location point to calculate the average.

In the probabilistic approach, instead of using the average value of RSS as a metric, a probability distribution of RSS is used [40, 41]. A fingerprint is modeled as a group of probability distributions of RSSs from detected APs. In the present case, the fingerprint can be defined as $(l, M)$ ,where $l$ denotes a location, and $M$ is a matrix containing RSSs that are used to model the probability distributions. In order to form a probability distribution, the RSS has to be measured multiple times. With multiple measurements the average and variance of the distribution can be calculated, assuming the RSS follows Gaussian distribution. The matrix $M$ is defined as:

$$M = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1a} \\ R_{21} & R_{22} & ... & R_{2a} \\ \vdots & \vdots & \ddots & \vdots \\ R_{k1} & R_{k2} & \cdots & R_{ka} \end{bmatrix}$$

Where $k$ is the amount of RSS for each AP, and $a$ is the number of detected APs.

In practise, there are many ways how to create the fingerprint radio map. First, the location area and the distance between fingerprints should be decided. Scattering the fingerprints uniformly throughout the area can possibly decrease positioning bias, compared to unequal distribution of fingerprints.

Next, we take a look at some possible examples of how to create the radio map. First example is simple and straighforward. The locations of fingerprints are decided and marked down to a map beforehand as in Fig. 7. Then with a device and software capable of measuring Wifi RSS, we physically walk to every position and stationary scan through all APs to measure their RSS, preferably multiple times to get an average. The red circles in Fig. 7 represent fingerprinting measurement locations. Because human body effects to RSSs, measuring in multiple orientations per location can give more accurate fingerprints. RSSs are then stored to memory with their location identifier. Aforementioned method is simple, but contains few drawbacks. First is that the method can be relatively time-consuming. Second, the method manages to create the RSSs fingerprints only in the predefined spots. Between the spots there is no measured evidence of RSS. This means the actual area where we do have empirical knowledge of RSS is usually small compared to the total area of the location. The density of RSS-knowledge in the area can be increased by raising the density of measurement spots, which in turn makes the method more time-consuming.

The second example utilises a graph for achieving a better coverage of the RSS information in the area [42]. Idea is to make a graph on top of the area map that is uniformly distributed as in Fig. 8. As in the figure, all the objects are connected through links. A user is then supposed to walk from object to object by following the link between them, and simultaneusly scan WiFi RSS with a device. When a user encounters object at the end of the link, he then marks down (with the aid of a software for example) the position of the object and the current time. This process is then repeated for each link in the figure, until the whole test area is covered.

In a WiFi RSS fingerprint map, the locations of RSS scannings should be known. In the above example, the locations of RSS scannings are not directly known. What

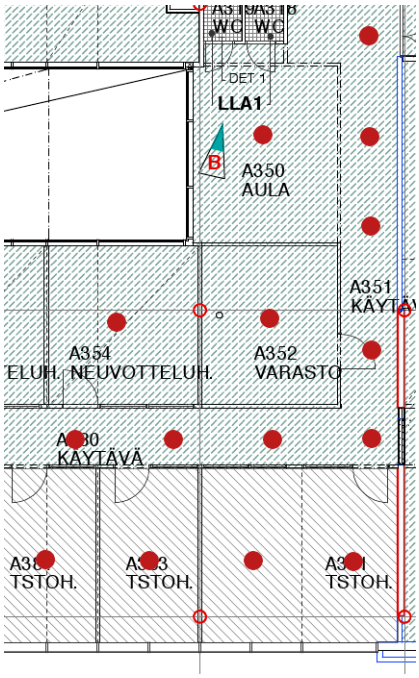


Figure 7: Fingerprinting; creation of a radio map with stationary measurements

is known in the example are: the departure and arrival times of the user traversing between objects, the times of WiFi scans, the locations of objects and the distance of the links between the objects. The RSS scan positions on a link can then be calculated by using the proportional relationship between time and distance. Let's have a simple example demonstrating this. Assume we have two objects A and B connected with a straight link. For the sake of simplicity, assume we only scanned once the RSS values at time $t_{scan}$, and the both objects A and B are located on the x-axis. What we are trying to calculate is the location where the RSS was scanned, denoted as $x_{scan}$. The coordinates of A and B are $(0, 0)$ and $(x_{dist}, 0)$. The user's time of departure from object A is denoted by $t_0$ and the time of arrival to object B is denoted by $t_{end}$. The time difference is then:

$$t_{diff} = t_{end} - t_0$$

Because of the proportional relationship between distance and time, the following relation holds:

$$\frac{x_{dist}}{x_{scan}} = \frac{t_{diff}}{t_{scan}}$$

From this we can easily solve the location of the scan $x_{scan}$:

$$x_{scan} = \frac{x_{dist} t_{scan}}{t_{diff}} = \frac{x_{dist} t_{scan}}{t_{end} - t_0}$$

The benefit of the graph-based method is that it gives more RSS-information about the area. One could also argue, that walking and scanning continuosly is

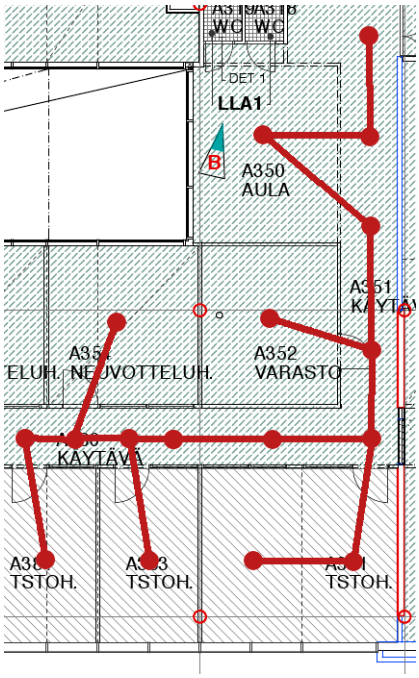


Figure 8: Fingerprinting; Graph-based method for creating the radio map.

faster than doing only stationary scans where you need to separately start and stop the scanning.

Both of the examples above are methods that use empirical measurements to create the fingerprint radio map. Another way to create the radio map is to analytically estimate the RSSs of APs by using different radio signal path loss models [43]. The analytical and empirical method can be used together in a beneficial way. By using the analytical method we can for example try to estimate the RSSs between measurement points. Secondly, the combination of these methods could result into a fewer required measurement points, which would make the calibration phase less time-consuming.

### 3.5.2 Online phase

In the online phase, a comparison is made between the online measurements and the fingerprint database created in the offline phase [41]. The position of a blind node is chosen based on this comparison. Many different comparison methods exists and in this thesis we will cover some of the common ones. Essential for almost all the comparison methods is that each RSS-vector measured in online phase is compared to a relevant group of fingerprints (sometimes to all fingerprints). One Wifi scan comprises of RSSs from all detected APs. The Wifi scan results are then compared with each relevant fingerprint. The RSSs of a Wifi scan are compared with fingerprint in such a way that each RSS is compared with the fingerprint's corresponding AP's data. The corresponding AP's data depends on the method used. In deterministic methods the corresponding AP's data for a fingerprint is usually one scalar RSS value. That value can be either one RSS measurement or an

average RSS from many measurements. In probabilistic methods the corresponding AP's data is a probability distribution of the RSS in that particular place. The probability distribution can be modelled with many measurements of RSSs in the spot. These give us the average and variance of RSS, which are needed for the Gaussian modelled probability distribution.

In deterministic-based methods the distance norm between the online RSS-vector and the fingerprint RSS-vector is calculated for all fingerprints. Distance norm can be for example Euclidean distance or Manhattan distance. Euclidean and Manhattan distance can be generalized by the Minkowski distance:

$$d_q = \left( \sum_{i=1}^{n} |s_i - S_i|^q \right)^{\frac{1}{q}}$$

Where $S_i$ is the RSS of the fingeprint, $s_i$ is the online measured RSS, n is the amount of APs detected, and $q$ determines the formula as Manhattan distance or Euclidean distance. If $q$ equals 2, then the distance norm is Euclidean distance. If $q$ equals 1, then the distance norm is Manhattan distance.

After all the distances between online measurement RSS-vector and fingerprints have been calculated, it is time to estimate the blind's position based on the distances. Each fingerprint is associated to a certain position. That means each distance is also associated to a certain position. In a *nearest neighbor method* [41], the location estimate is obtained by choosing the position that has the minimum distance compared to other position's distances. In a *K-nearest neighbors* (KNN) [44] method, k positions having the smallest distance are chosen instead of just one. The location estimate is then obtained by averaging the k-nearest neighbours' locations. Modification of the KNN method is the so-called WkNN method (Weighted k nearest neighbors). WkNN works similarily to KNN mehtod, except after finding the k best matching measurements it assigns weights for each of these measurements. The weights can be for example, the inverse of the distance.

In the probabilistic fingerprinting, signal measurements are considered as samples from a random variable. The random variable follows a certain probability distribution. The probability distribution can be either a histogram-based or parametric.[31] The histogram-based probability distribution models the signal variations by using a histogram which consist of previous observations. The parametric probability distribution model can be for example Gaussian distribution, that has a probability density function (pdf) as:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

Where $\mu$ is the mean value of the random variable, $\sigma^2$ is the variance, and $x$ is the observed value.

What probabilistic fingerprinting tries to find is the most probable position of the blind node. Let $l$ denote the location variable. The idea is to compute the conditional posterior pdf for each relevant fingerprint, and select the position that has the highest probability value.[41] The posterior distribution means that what

is the probability that a blind node is at location $l$, when we have received an observation vector $x$, containing the received RSS from the detected APs. From Bayes rule we can get the posterior distribution:

$$p(l \mid x) = \frac{p(x \mid l)\, p(l)}{p(x)}$$

Where $p(l)$ is the prior probability of the blind being at location $l$ without knowing the value of observation vector $x$. The prior distribution can be used to include background information, for example user profiles. For simplicity we use uniform priors that gives no bias towards any location. The $p(x)$ is treated as a normalizing constant if needed, since it does not depend on the location variable $l$.

The part we are most interested in is the $p(x \mid l)$, which is called the *likelihood function*. It gives the probability that we have observed $x$, when the blind is located at $l$. The likelihood function can be implemented in multiple different ways. In this thesis we will cover two different types of methods how to implement the likelihood function, the *kernel method* and the *log-sum method*.

The operating principle of the kernel method is that after we have observed RSS vector $x$, we calculate probability for each RSS by using kernel function. Then we sum up those probabilities, and divide the sum by the amount of RSS values in vector $x$. More formally:

$$p(x \mid l) = \frac{1}{n} \sum_{i=1}^{n} K(x; x_i)$$

Where $K(x; x_i)$ is denoted as the kernel function. Kernel function can be for example the Gaussian probability density function presented in equ. 4. To clarify things little, we have to remember that each fingerprint in a database consists of training vectors $x_i$. Each of these training vectors model a particular AP's RSS probability distribution in that location. [31]

The log-sum method works a little differently. If we assume that the APs are independent on each other, then the likelihood function can be calculated by first estimating the probability of each RSS, and then multiplying those probabilities together. The probability can be estimated for example by using the Gaussian probability density function presented in equ. 4. If one of the probabilities is a value near zero, then it can affect a lot to the end result, because the probabilities are multiplied together. In order to prevent this kind of underflow, the likelihood function is calculated on a logarithmic scale. In logarithmic scale the product of probabilities converts into a sum of logarithmic probabilities:

$$p(x \mid l) = \prod_{i=1}^{N} p(x_i \mid \mu_i, \sigma_i) \leftrightarrow \log p(x \mid l) = \sum_{i=1}^{N} \log p(x_i \mid \mu_i, \sigma_i)$$

Once the likelihood function has been calculated to each relevant location, the position with the highest probability can be chosen as the position estimate. Or the weighted average of the k-highest ranking positions can be chosen. In probabilistic approach the offline measurements can be grouped into a different clusters.

Each cluster can then represent as one location. This means that inside the cluster there are multiple fingerprints, each to be considered as the same location. In the aforesaid setting the fingerprints can be for example combined as one fingerprint. Grouping the offline measurements into clusters and using the kernel method is in the experiments of *Roos*[41] produced worse results than the kernel method without the clustering.

Next we will consider methods that try to increase the accuracy of the positioning by using the history position estimates to filter out the false position candidates for the next location. Assume our positioning system updates the location of the blind node with continuous interval of one second for example. Can we, by some way, use the history locations in the determination of the next location? The use of filtering in indoor positioning is discussed on the next section.

## 3.6 Filtering

Filtering is a technique that can be used in indoor and outdoor positioning to gain more positioning accuracy. Filtering can be utilised on almost any indoor positioning technique, where locations are updated continuosly with particular time interval. Basic idea of filtering is to filter out false position candidates based on previous history estimates. For example, if a position is updated every second, then we can assume, that the next position cannot be far away from the previous one. With this in mind, we can filter out all positions that are too far away from the previous position. But what if the previous location estimate was wrong, and the real position was relatively far away from it? Are we then mistakenly actually filtering out positions that could be right? For example, in fingerprinting technique, different places can share a very similar fingerprint, that could lead to a mistake in estimation. As it turns out, filtering requires more complex methods to provide robust results. In this subsection, we will review two different filtering techniques that can enhance the accuracy of a positioning system. The first technique is denoted as a simple filtering with error estimate.

### 3.6.1 Simple filtering with error estimate

One solution to the aforementioned filtering situation is to estimate a positioning error. If a location estimate receives a high error-level, then that estimate would not be used in the filtering. Only the location estimates with low error-level would be utilized in the filtering. For the fingerprinting-based positioning, the error estimate can be calculated for example the following way[45]:

$$E\left(l_e \mid x\right) = \sum_{i=0} p\left(l_i \mid x\right) dist\left(l_i, l_e\right)$$

Wherein:

- $l_e$ is the estimated location for which the error estimate is being calculated,
- $x$ is the RSS-vector observed at the observation point,

- $l_i$ is the location variable,
- $p\left(l_i \mid x\right)$is the posterior (probability) of the location $l_i$,
- $dist\left(l_i, l_e\right)$ is a distance function that calculates distance between $l_i$ and $l_e$.

The error estimate for the estimated location is calculated by a sum of products. The sum consist of each different position's probability of being the correct position, multiplied by the position's distance to the estimated location. Based on my personal experiences, the simple filtering with error-estimate can increase a positioning accuracy of a positioning system.

The simple filtering with error-estimate can improve the positioning accuracy, but it can be improved even more with complex filtering. The simple filtering fails at some situations where a more complex filtering would not. Let's have an example. In our example, we have WiFi fingerprinting-based indoor positioning system that applies the simple filtering with an error-estimate. Assume we have a situation as in Fig. 9, where a blind node is first at location A. Then the blind node (person with a device) is moving linearly with a constant speed to location B, and from that to location C. The speed remains constant whole time during the movement, and the location of the blind is estimated at constant interval of 3 seconds, for example. First, estimation of the location is done at point A, second estimation is done at point B, and third estimation is done at point C. The movement of the blind can then be described as:

$$A \rightarrow B \rightarrow C$$

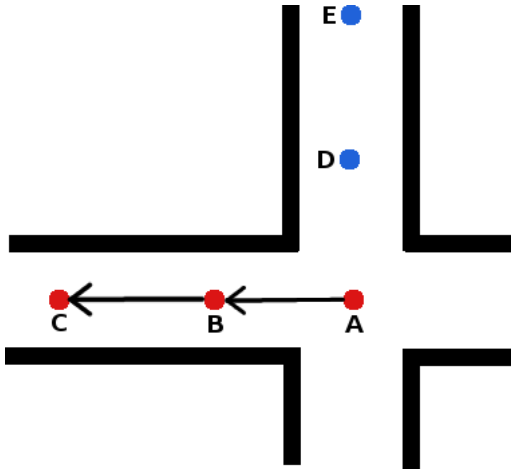


Figure 9: Example of filtering error in simple filtering method

The estimation of the location of the blind however is not correct. The location of the blind was estimated first to A, then to B, and last, it jumped to E. The positioning estimates of the system about the location of blind can then be desrcibed as:

$$A \rightarrow B \rightarrow E$$

Why in our example did the estimate jump from B to E, even the correct positions were A, B and then C? Let's go through step by step, what could have caused the jump, and why didn't the filtering work.

Step A:

We assumed that the system was started when the blind was located at location A. No previous history positionings existed at this point. The blind scanned RSSs of APs, send the results to a server, which then run a fingerprinting algorithm, that estimated the users location to be A. Location A had most similar RSS-readings with the scanned online measurements. The server also calculated the error-estimate for the location, and the error-estimate was good for the position. It means, that the location would be used in filtering for the next position estimates.

Step B:

At step B, the blind arrives to location B, and sends new measured RSS again to the server for location estimation. The server estimates the location by using fingerprinting and filtering techniques. The fingerprint in B is the closest match with the current online RSS that were scanned. The position D has a bad probability, so it is not selected. The server estimates user position as B. But, the error estimate for location B was bad. It means that the server cannot say with high certainty, that the users correct location is B. Therefore, this location estimate of B is not used in the future for filtering purposes.

Step C:

The blind arrives to location C, and sends again new measured RSSs to the server, which estimates the position. This time, the location E shares a similar signal environment with the location C. The server uses fingerprinting algorithm to estimate the position. The result of the estimation is, that location E has the most similar fingerprint with the current scanned RSS. After this, the server applies filtering to the results. The only history position that could now be used for filtering is the position A, since it is the only one that had a good error-estimate. The position A is as far away from the position C and the position E, so the server cannot filter out position E, and therefore it chooses E to be the most correct position.

Is there anything in this situation that could have been done to prevent the faulty jump from B to E? What the simple filtering technique doesn't take into account, is a path probability. In order for the LD to get from A to E, it would need to go through D. At step C, the system could take into account, that the probability of the position D was bad at step B. This information could then be used in filtering, which could have made the difference between places C and E. Next, we will review particle filter and its use in indoor positioning to overcome demerits of simple filtering.

### 3.6.2 Particle filter

Using a particle filter in indoor positioning [46, 47] can unravel the problems of the simple filtering described before. A Particle filter is an implementation of the Bayes filter algorithm. Particle filter belongs to the family of the sequential Monte Carlo methods. The idea of a particle filter is to maintain a probability distribution of the location estimate by a set of initially randomly selected particles assigned with weights:

$$Bel\left(s_t\right) = \left\{s_t^i, w_t^i\right\}$$

Where:

- $Bel\left(s_t\right)$is the belief that the system is at state $s_t$,
- $t$ is the time varibale,
- $s_t^i$ is a discrete hypothesis about the location of the object at time $t$,
- $w_t^i$ is a weight value of the corresponding particle.

The weights are called *importance factors*, and they are non-negative values. The sum of all the particles' weights equals as one.

Lets overview the process of the particle filter in the context of indoor positioning and this thesis, using a fingerprinting-based positioning as the base positioning method. The process starts when a blind node makes a new measurements of the APs' signal strengths. This leads to an update of the particles' location and weight through a procedure named Sequential Importance Sample with Resampling (SISR). First, the particles' places are predicted using a *motion model.* After this prediction, the particles are weighted according to the sensors' *likelihood model* of the new measurement. Final step is to resample the particles using importance sampling. The resampling of particles means that the particles are replicated in relation to their weights. The particles with higher weights will be replicated more, and the particles with lower weights will be either deleted or replicated less. The resampling results to a set of particles with equal weights. A method called KLD adaptation (*Kullback-Leibler Distance*) can be used in each step to obtain the optimal number of needed particles [48].

The prediction of the particles' places using a motion model is an implementation of the Bayes filter prediction step. What it means is that we have a certain motion model for each particle, and based on the motion model and time elapsed from last measurements, we make a hypothesis about the location. The motion model depends on the object we are locating and it can be created and decided in many ways. If the blind node uses IMU-sensors to measure its movement, then those sensor measurements can be used to infer the motion of the LD. If there is no access to any sensors and the blind nodes are humans, then a general human motion model can be applied for the particles. A general human motion model consist of the human's speed and direction. This means that in addition to the weight, each particle would also have a speed and a direction in its belief state. The speed and direction of the

particles are resampled in each iteration of the SISR procedure. This can be done for example by modelling the speed and direction as Gaussian probability distributions and drawing new values from these distributions at each iteration. After the motion variables are updated, the particles are moved to the positions according to the their motion model. The motion model can be also constrained by a collision detection algorithm in a way that particles cannot go through walls in the map.

The next step in SISR is the weighting of particles according to the sensors' likelihood model. The weighting and resampling of the particles is an implementation of the Bayes filter update step. In the case of fingerprinting-based positioning, the likelihood model corresponds to the likelihood function $p(x \mid l)$, presented in subsub-section 3.5.2 . Where $x$ denotes the measurement vector, and $l$ is the location variable. Initially, equal weights can be assigned to all particles. Once new measurements occurs, the old weights are updated via:

$$w_t^i = \eta \cdot w_{t-1}^i \cdot p(x \mid l)$$

Where $w_{t-1}^i$ is the particle's previous weight, and $\eta$ is a normalizing constant. The normalizing constant $\eta$ is a constant that normalizes the sum of the new weights to equal as one: $\sum_{i=1}^{n} w_t^i = 1$.

Final step in the algorithm is the resampling of the particles. The particles can be resampled when the filter has become degenerated. A degenerated filter is a filter where all but one importance weights are close to zero. By estimating the effective number of particles [49], we can determine whether there is a need for resampling:

$$N_{EFF} = \frac{1}{\sum_j \left(w_k^j\right)^2}$$

If $N_{EFF} < N_{treshold}$, then there is a need for resampling. The state of the system can be estimated by calculating a weighted average of the particles position [50]:

$$s_t = \sum_{i=1}^{n} w_t^i s_t^i$$

By repeating the algorithm's processes over time, the location of the LD can be estimated recursively.

# 4 Indoor Propagation

The main invention of the thesis is closely related to a radio magnetic signal propagation indoors. Therefore, we will review the most relevant characteristics and propagation models of radio signals indoors. In order to understand and model the indoor propagation of radio signals, it is necessary to apprehend the basic mechanisms of wave propagation. After introducing the basic propagation mechanisms, characteristics of a wireless channel will be outlined. These include fading in terms of *shadowing. multipath fading*, and average path loss. The average path loss can be modelled with different deterministic and empirical models. Some of the indoor propagation models will be introduced in the last subsection of this section. Modelling indoor propagation can be challenging because an indoor environment can be very dynamic and heterogeneous, with movement and a lot of different obstacles, floors, walls and humans.

## 4.1 Basic radio wave propagation mechanisms

When a radio wave propagates in the troposphere and encounters obstacles, it is affected by different kind of phenomens. These include signal reflection, refraction, scattering and diffraction. All the aforementioned mechanisms can cause signal to traverse multipath between sender and receiver. Multipath issues can cause problems for different positioning techniques. In this subsection, we will examine these basic mechanisms starting from reflecion and refraction.

### 4.1.1 Reflection and refraction

Reflection and refraction are phenomens that occur when a radio wave impacts an object that is larger in size than the wave's wavelength [51]. More specifically, we are considering what happens when a radio wave hits to a boundary between two different mediums as in Fig. 10. According to *Maxwell's equations*, two new waves are formed, and those waves have the same frequency as the origin wave [32]. The first wave reflects back to medium 1, and its reflection angle $\theta_r$ is the same as the angle of incidence $\theta_i$ of the origin wave. If the surface of the mediums is not smooth, but more rougher, then the reflected wave will become scattered to a more broader area. The relation between the incidence angle and the reflection angle is the *Snell's law of reflection.* The second wave absorbs into medium 2 with angle $\theta_{refract}$ and is hence denoted as the refracted wave in this thesis. The departure angle of the refracted radio wave $\theta_{refract}$ can be solved using the *Snell's law of refraction*:

$$\frac{\sin\theta_i}{\sin\theta_{refract}} = \frac{k_{refract}}{k_i}$$

Where $k_{refract}$and $k_i$ are the refractive indices of the respective mediums. This equation only holds in lossless media. In practise, all mediums are lossy. For lossy medium, there is an extansion for the *Snell's law of refraction.* More details of the extension can be found in [52].

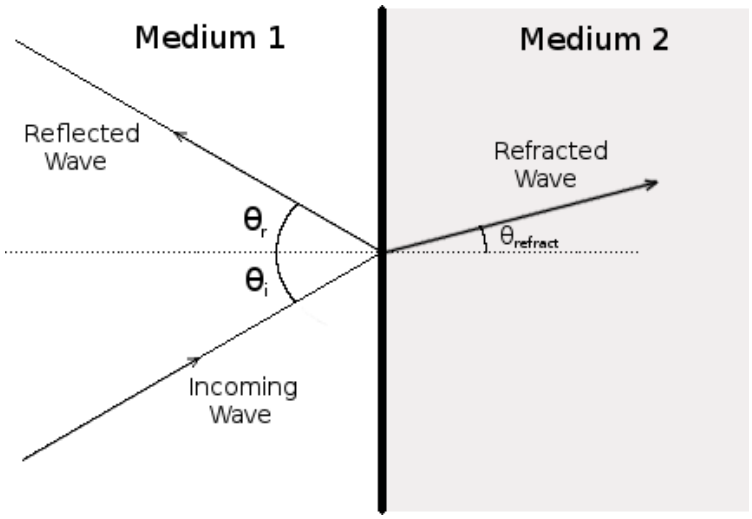


Figure 10: Reflection and refraction of a radio wave

### 4.1.2 Diffraction

The diffraction of a radio signal is a phenomenom that is related to a shadow region, that is located behind an object.In shador region, there is no direct LOS (line-of-sight) from the radiator [32]. The diffraction phenomen occurs at the edge of an object, and causes radio field behind the object where there are no LOS [51]. The diffraction can be best understood by using *Huygen's principle*. *Huygen's principle* states that each element of a wavefront can be regarded as an independent spot radiator by itself. Each of these independent spot radiators produces spherical wavelets. In a normal wavefront, wave vectors that are not in the direction of the wavefront's propagation are negated, because there is always a wave vector of opposite direction. At the edge of an object, these negations will not occur.

Let's have an example using a perfectly absorbing screen and a radio wave that is propagating towards it as in Fig. 11. The perfectly absorbing screen will not pass through any waves that reach it. Each wavefront is full of individual radiator elements. When a wavefront reaches the absorbing screen, the element radiator that is closest to the edge will radiate its wave also to the shadow region. The diffracted signals will have smaller signal strength than the original had.[32]

## 4.2 Shadowing and multipath fading

Shadowing is a phenomen where a path loss of a radio signal varies depending on the path of the signal [32]. Even if the distance of the paths would be equal, some paths will suffer more path loss, where others will suffer less, depending on the particular clutter (trees, buildings, cars for example) of the path. Let's have an example where we move a mobile phone around a base station (BS) at a constant distance as in Fig. 12. The signal path between mobile and BS depends on where the mobile is located. In practise, the path can vary a lot. One path can consist of for example multiple block of flats, where another path can consist of a direct line-of-sight between mobile

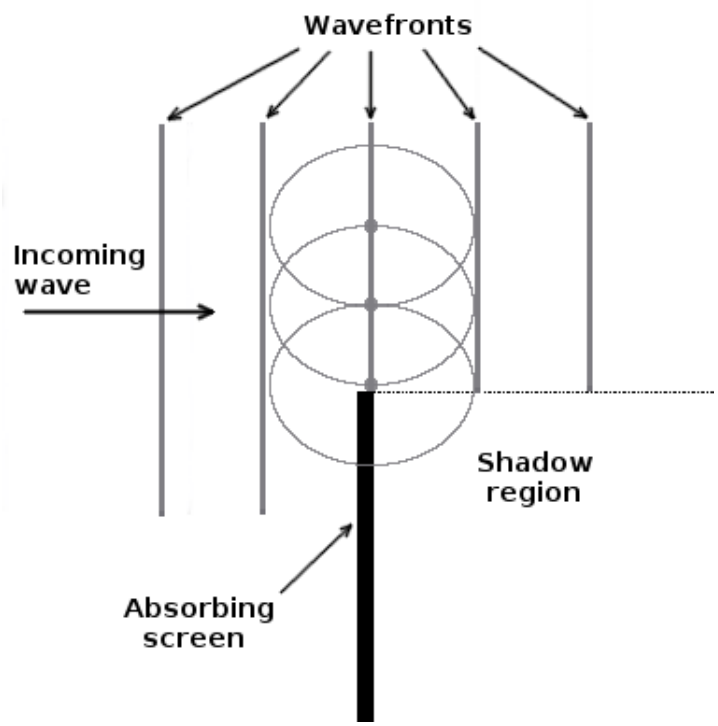


Figure 11: Example of diffraction caused by an absorbing plane

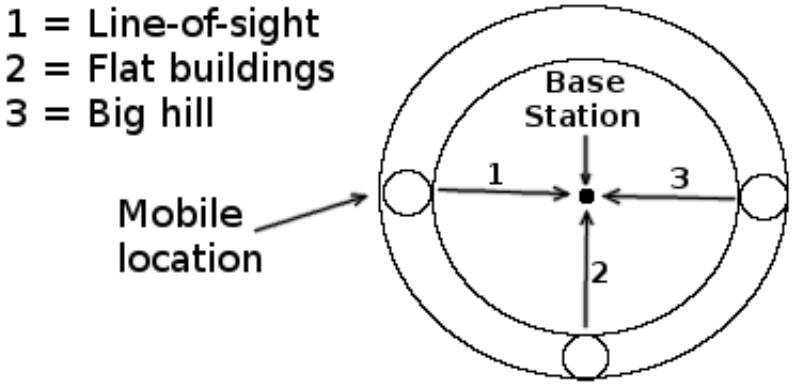


Figure 12: Different fixed range path profiles

and BS. The difference in path loss between these two paths can be remarkable. This effect is called *Shadowing*. If we form a probability distribution of the local mean RSSs around the BS, it follows approximately log-normal distribution. Log-normal distribution means, that the received signal strengths measured in decibels have a normal distribution. Shadowing can occur not only in outdoor environment, but

also indoors.

A radio signal can propagate through multiple paths to the receiver antenna. These co-signals can arrive at different phase, and when they coincide at the receiver antenna, the signal strength can be either abolished or increased. The received signal is a sum of many waves with random phases. This phenomen is called multipath fading and it is modelled for example by Rician or Rayleigh distribution. Multipath fading can be caused by the formerly introduced mechanisms: reflection, refraction, scattering and diffraction.

## 4.3 Indoor Propagation Models

A wireless channel can be characterized by fading and average path loss. Fading was reviewed in the previous subsection, and the meaning of a signal path loss was given in section 3.4 (RSSI-based ranging). The average path loss of a radio signal is distance dependent, and it is modelled with deterministic and empirical models. The free space path loss model, that was introduced in section 3.4, is generally seen as inadequate for modelling the indoor path loss. This is because indoor environment is no free space. Indoors, the walls, floors, and moving humans cause many phenomens to a radio signal, which are either hard or impossible to model correctly. The materials inside buildings will attenuate the signal strength more than the attenuation of a free space would. In this section we will review various indoor propagation models starting from Keenan model.

### 4.3.1 Empirical path loss models

The Keenan model [53] tries to take into account the path loss that comes from different floors and walls inside a building. The basis is the same as in the free space model presented in section 3.4, but in addition, there are loss factors for walls and floors that reside between the receiver and transmitter. The formula for Keenan model is:

$$L = L_1 + 20 \log r + n_f a_f + n_w a_w$$

Where $L_1$ is the loss at $r = 1$ m, $n_f$ and $n_w$ are the number of floors and walls, r is the distance between receiver and transmitter, and $a_f$ and $a_w$ are the attenuation factors in decibels for each floor and wall.

ITU-R model is similar to Keenan model, except it only accounts the floor loss explicitly. The loss between points on the same floor will be calculated by changing the path loss exponent. Thus, the ITU-R model's formula is:

$$L_T = 20 \log f_c + 10n \log r + L_f\left(n_f\right) - 28$$

Where $f_c$ is the frequency of the signal in megahertz, $n$ is the distance path loss exponent, $r$ is the distance between receiver and transmitter, and $L_f\left(n_f\right)$ is the floor penetration loss factor, which depends on the floor type and number of penetrated floors $n_f$.

Keenan and ITU-R model were empirical path loss models. Creation of an empirical path loss model consist of making many path loss measurements, fitting an appropriate function to the measurements along with multiple different parameters. The empirical models suffer from various disadvantages:

- They require a use of spesific parameter range that was used in the original measurement set.
- They don't take into account the mechanisms by which propagation occurs
- Environment classifications in empirical methods are subjective, which means their meaning can change in different context.

### 4.3.2 Physical path loss models

Physical path loss models strive to model the path loss by understanding the physical mechanisms of propagation [32]. These include geometrical optics and the uniform theory of diffraction (UTD). The geometrical optics refer to the basic ray propagation mechanisms with diffraction excluded. The process of using geometrical optics in order to find an approximate value for the electromagnetic field is following: First, all possible ray paths between the tranceiver and receiver are calculated based on the Snell's laws of refraction and reflection. Then the Fresnel reflection and transmission coefficients are calculated for each reflection and transmission point. Third, the amplitudes for each ray paths should be corrected by taking into account the wavefront curvature from the source and the curvature of any boundaries. Then finally, all ray paths are summed at the end point. The last step requires that the different amplitudes and phases are taken into account. The geometrical optics does not take into account the diffraction phenomen that was described earlier. Therefore physical models usually tries to combine the geometrical optics and the UTD in modelling of path loss.

Ray tracing is a method that applies the geometrical optics and UTD theory in modelling a path loss[32, 54]. In Ray tracing, all the possible ray paths between the transmitter and receiver are calculated. In indoor environment, ray tracing can be computationally very time-consuming. This is because indoors can contain lot of surfaces between mediums that causes reflections, refractions and diffraction. One should know all the parameters regarding to the indoor environment, materials and thickness of walls and floors, very accurate blueprints, and transmission related parameters, in order to get reliable results.

Ray launching is a similar method to ray tracing, except instead of trying to find all possible ray paths between a transceiver and a receiver, test rays are send at a amount of discrete azimuths from the transceiver [54]. The path of each test ray is calculated based on the geometrical optics and the UTD theory. The propagation of a ray is terminated when its power level decreases below preset threshold value. Paths to the estimated point are detected by either using a distant dependent reception sphere, or angular information at the image signal (caused by reflection).

# 5 Ekahau Positioning System Architechture

This section describes the architechture behind Ekahau's positioning framework [55], that is used as a base positioning system in the tests done in this thesis work. The Ekahau positioning system uses WLAN and optionally infrared sensors for localization. More specifically, it applies Wifi fingerprinting method, that was described in the section 3.5. The system locates mainly tags manufactured by the Ekahau company, but there are also smartphone clients for the system. The Ekahau positioning system requires full coverage of WiFi APs for the area of interest (in positioning) in order to work. The Ekahau's positioning framework consists of 4 main components, which together form a real-time locating system (RTLS):

- The **E**kahau **S**ite **S**urvey (ESS)
- The **E**kahau **R**eal-Time Location System **C**ontroller (ERC)
- The Ekahau Vision
- The located tags

In the context of indoor positioning, the ESS is a software implementation tool for the offline-phase of the fingerprinting method. ESS can be used to create a model of the location area's radio map, consisting all Wifi AP information with RSS (Received Signal Strength). The ERC-controller is a server program, that implements positioning algorithms and makes all positioning estimates. It is required, that a radio fingerprint map is created for the area of interest, before ERC can be used for positioning. Therefore, the model created by ESS is first loaded to ERC. This model acts as the fingerprint radio map, that was described in the section 3.5.1. The Ekahau Vision is a tool that provides a visual interface for tracking different clients and obtaining report-data of different events that may have occured. The Ekahau Vision was not used in this thesis, and therefore it will not be reviewed any further.

The Ekahau positioning system can track three different types of tags. Each of these tags applies to WiFi standards and have a Network Interface Controller (NIC), that is used to connect the tags to a network. Next, let's review the three main components of the Ekahau positioning system separately, to have a better understanding of them.

## 5.1 Ekahau Site Survey

The Ekahau Site Survey (ESS) is a Windows application for site surveying Wifi networks. Its interface is shown in Fig.13. ESS can be used to measure and estimate various attributes of WiFi networks. These include but are not limited to: Data rate, network health, signal strength, number of APs, channel overlap and signal to noise ratio. For the context of indoor positioning, the most important feature of ESS is the ability to create fingerprint radio maps. A general description of a fingerprint radio map was given in section 3.5. Creating a fingerpint radio map requires performing a site survey with the tool. In a site survey, the application

scans WiFi networks and simultaneusly maps the results into a provided map of the location. The map of the location has to be uploaded to the ESS tool before site surveying is possible. The device that is used for scanning the WiFi network can be configured manually.

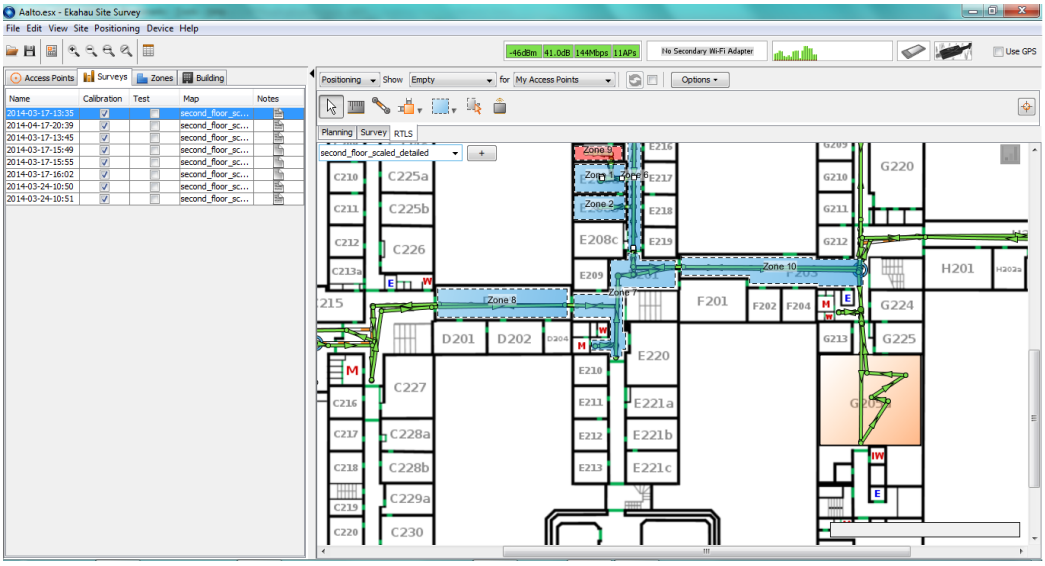


Figure 13: The Ekahau Site Survey

Let's focus more on the process of making a site survey with ESS. The process requires that user have to walk with a laptop running ESS, and pinpoint simultaneusly to ESS where he is located on the map. In ESS, the creation of a fingerprinting radio map is done by using continuous WiFi scanning, and interpolating the scan results to a straight line between two points, the begin and end point. This method was described in section 3.5.1, and it is also used in war-driving and war-walking methods [42]. A site survey is like a graph, consisting perpendicular links between points. The basic usage is, that a user marks the beginning point of a link and starts walking perpendicularly with constant speed. When he reaches the desired end point of a link, he marks the end point to the map. Everytime when a new scan is completed, a timestamp of the scan is collected. Timestamps are also collected when user marks starting and ending point of a link. When ESS has the times of scans, times and coordinates of a startpoint and an endpoint, it can interpolate the scan results along the link.

ESS stores surveys to a model file with .esx extension. A model file contains all surveys that have been made, maps used in the surveys and results of the surveys. In order to make the model more optimized for indoor positioning, there are few tasks that should be made. First is a pixel-to-meter scale. In ESS, there is a way to set a pixel-to-meter scale, that indicates how many pixels is one meter. The scale is used by the positioning algorithms to enhance positioning with filtering. It will be discussed more in the further sub-section. Second feature that can improve positioning accuracy is the utilization of rails. In ESS, the user can draw straight lined rails to the map, that indicates how the located device(s) can move. This

decreases the probability, that user could move through walls. The following example demonstrates how rails can benefit positioning. In Fig.14, there are two rooms with rails connecting them. The rails indicate how a located object can move. Let's assume we have estimated users position to location A. In order for the user to be positioned in location B, it would have to go along the rails instead of through the wall. The last improvement for positioning is zones. Defining areas to a reasonable zones will affect the positioning algorithm's particle filter to perform better. In ESS the user can easily draw zones of areas. A common zone is a one room, for example.

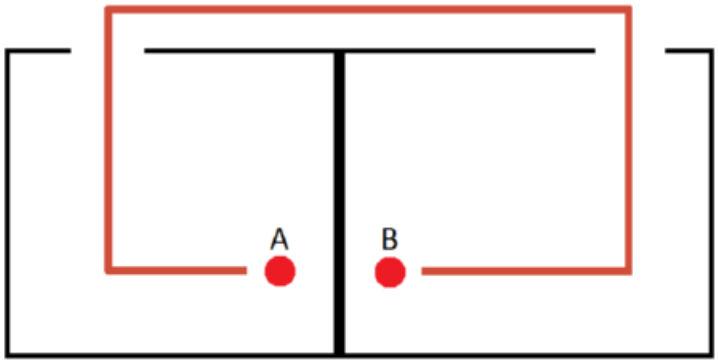


Figure 14: A rail example

## 5.2 Ekahau Real-Time Location System Controller (ERC)

The Ekahau Real-Time Location System Controller (ERC) is a server application, that makes all the positioning estimates and keeps tracks of all the locateable devices connected to the Ekahau system. ERC has a web interface, which is shown in Fig.15. ERC provides an API, through which other applications can get various informations about tracked devices. In order for a device to be located by the Ekahau system, it is required to be connected to ERC. Connecting an Ekahau tag to the server can be made by a Windows software called Ekahau Tag Activator, which comes bundled with the ESS installation. Once the tag is properly configured, it begins the scanning of WiFi networks and sends continuously results to ERC.

For positioning, ERC uses a fingerprinting-based positioning algorithm combined with a particle filter. As a fingerprint radio map, ERC uses the model produced by ESS. User has to upload the model to ERC manually through a web interface. Once the model is uploaded, and ERC receives online measurements of blind nodes, it can estimate their positions. It compares the online measurements to the fingerprint radio map, and determines the new position estimates with the aid of a particle filter. The fingerprinting technique and particle filter were presented in sections 3.5 and 3.6.2. In ESS, a user can choose a pixel-to-meter ratio for the map. This ratio is used by ERC's particle filter. Moving the particles on the map according to their physical model requires that the pixel-to-meter ratio is known.

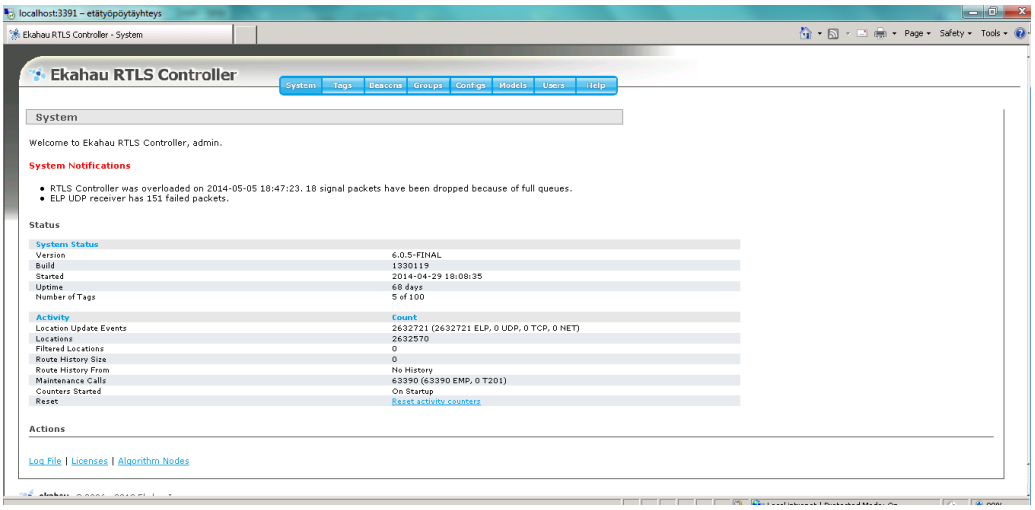


Figure 15: The Ekahau Real-Time Location System Controller web interface

## 5.3 Ekahau Tags

The Ekahau positioning system can locate different types of tags, manufactured by Ekahau themself. The tags are small and battery powered. The tags can be recharged by using Ekahau manufactured recharger component. Currently existing tags are: A4, A4+, B4 and W4. The tags communicate with ERC server through WiFi WiFi 802.11 APs. By using the ERC web interface, the configurations of the tags can be changed. The basic functionality of a tag is to perform WiFi network scanning according to their configuration, and then sending the scan results to ERC server. In order for a tag to perform its basic functionality, it has to be connected to ERC server. The connection can be established as described in the previous subsection. The tags have a motion detector, and its data is utilized by the positioning algorithms for a better positioning accuracy.


Figure 16: A4 Asset Tag [56]

Let's review the different tags next. First tag to be introduced is A4 [56]. The main purpose of the A4 asset tag (seen in Fig.16) is to enable asset tracking by attaching it to an asset. Here are some of the most interesting attributes of A4 tag:

- The battery life of A4 tag is up to five years.
- The size of A4 is small. Its dimensions are: 1.8 x 2.2 x 0.7 in / 45 x 55 x 19 mm.
- A4 contains two programmable buttons for alerts or other call purposes.
- It is possible to remotely activate audio alerts in A4, as it contains small speaker.


Figure 17: Ekahau B4 Staff Badge Tag [57]

Figure 18: W4 Wristband Tag [58]

B4 badge tag [57] is a tag that will be used in the tests in this thesis. The B4 tags (Fig. 17) are meant to be used for staff or visitor tracking for various purposes. A common use case is tracking of staff members for the sake of their safety. The B4 tags have a size of a typical ID-card, and can be worn around a neck on a lanyard. Attributes of B4 are:

- Safety switch for lanyard. When pulled, an alert is send to a desired (programmed) location.
- Programmable menu button and two call buttons.
- Environment-friendly re-chargable batteries.

- Ability for bi-directional texting communication.
- An organic LED display that shows all the messages.
- Possible to launch audio alerts.
- Dimensions: 2.36 x 3.54 x 0.33 in / 60 x 90 x 8.5 mm.
- Weight: 1.7 oz/ 46 g.

W4 wristband tag is an alternative to B4 tag for person tracking. The main difference is, that it can be weared as a wristband as seen in Fig.18. More details can be found at Ekahau W4 wristband datasheet [58].

# 6 Test scenarios and results

In this section we will present all the test scenarios made and show their results. Two different kind of test scenarios have been conducted. In the first test scenario, the RSS of a static emulated WiFi AP was measured in pretedermined places. The aim was to see if there is a significant difference in the RSS between different rooms, and that how much did the walls attenuate the signals. In the second test scenario, the Ekahau positioning system was tested with and without the aid of a static BLE beacon, and then the room accuracy of the test area was analyzed between the scenarios.

## 6.1 RSS measurements

In this subsection, we will go through the RSS measurement test and show its results. The main components of the test were: a one static emulated Wi-Fi AP and two Ekahau B4 tags. The static Wi-Fi AP was placed in one of the rooms at Ekahau office. Then it was configured to send Wi-Fi beacon frames at certain interval. With the B4 tags, the RSS of the beacon frames was measured. During the tests a test person weared the two B4 tags, one in front of him, and one in behind. Possible attenuation caused by human body could then be measured. Tests were made with different output gain levels of the emulated Wi-Fi beacon. What we were mainly interested to observe was, that whether there was a significant difference in the RSS between the room where AP was located, and its neighboring rooms. The reason why we were interested in this is because, if there is a noteworthy difference, then that would indicate higher chances of improving the room-level accuracy of the system by using a static beacon (AP or a BLE beacon) to make fingerprints more unique. We were also interested to see how different output gains of the emulated Wi-Fi AP affect to the RSS measurements.

### 6.1.1 Methodology of the RSS measurements

We determined static measurement places where measurements would be made. A map indicating all the measurement places can be seen in Fig. 19. In the Fig. 19, numbers indicate the places of measurements. Total number of measurement places is thus 14. The emulated Wi-Fi antenna was placed at the position marked with a capital A letter.

Let's review the equipment used in this test. As mentioned before, two Ekahau B4 tags were used to measure the RSS of an emulated Wi-Fi AP. The Wi-Fi AP was emulated by using a USB-connected wireless Wi-Fi network-adaptor. This adaptor was plugged into a laptop, and then configured with a suitable program to broadcast Wi-Fi beacons with regular interval. Because of the possible interferences caused by the integrated laptop, we configured the Wi-Fi adaptor to use an omnidirectional external antenna that was connected to it through RP-SMA (Reverse Polarity Sub-Miniature version A) connector and a coaxial extension cable. The omnidirectional antenna and its placement in the test room can be seen in Fig. 20.

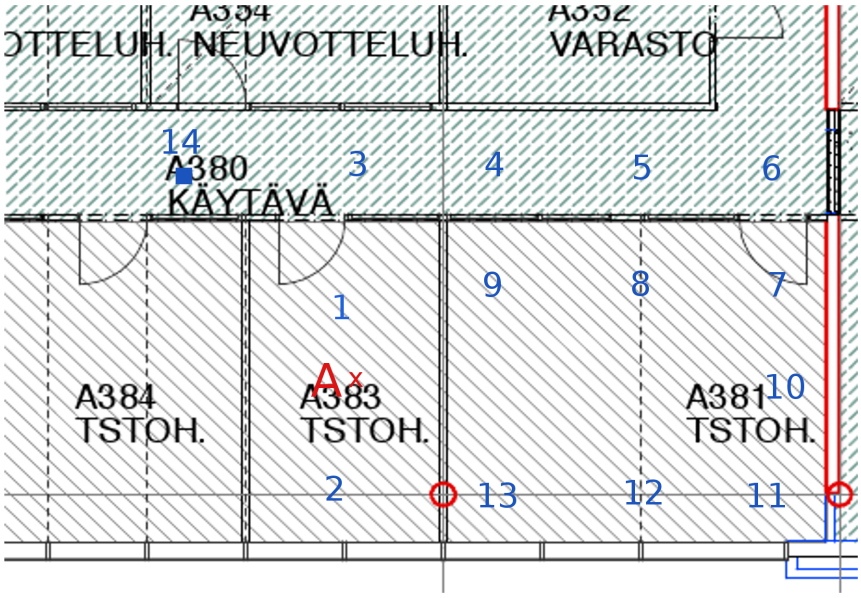


Figure 19: The map of measurement locations at Ekahau office

Figure 20: The omnidirectional antenna and its placement during the test

The USB-based network adapter was found to be leaking in the test. Therefore we made a following rearrangement: we placed the laptop with the USB-adaptor inside a metallic box which behaved as a Faraday's box, thus not letting any radio signals out. The metallic box contained an interface for extending our antenna outside of it.

As mentioned earlier, we were also interested to see how different output gains of the emulated Wi-Fi AP would affect to the RSS measurements. In order to get altered output gains, we connected the antenna cable with an attenuator capable of adding extra attenuation to the signal. A shematic diagram of this measurement setup can be seen in Fig. 21.

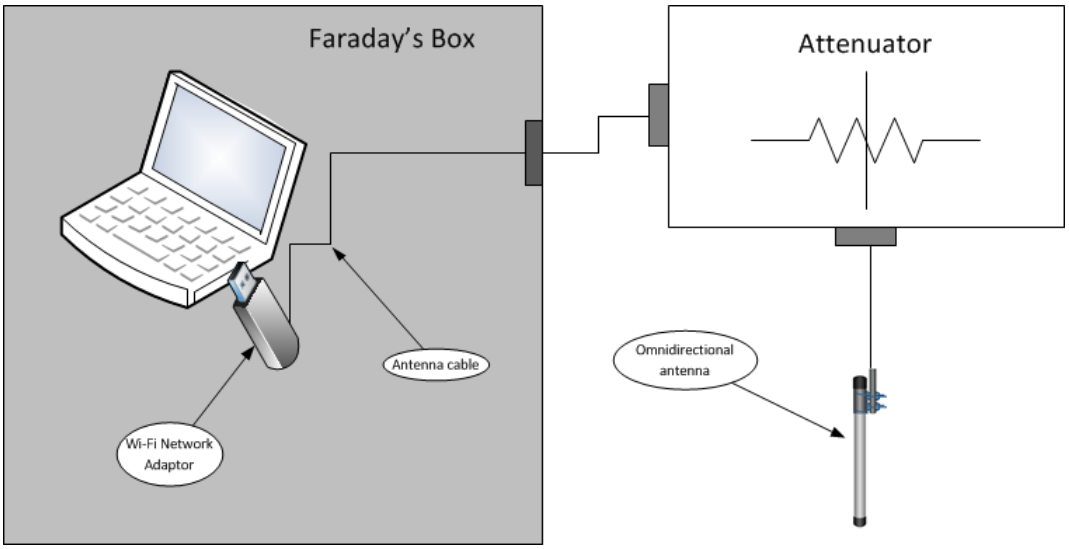


Figure 21: The test setup for emulated Wi-Fi AP.

Next, let's review details of the measurement process and what statistical attributes where calculated from the RSS and how. The emulated Wi-Fi AP was configured to continuously send Wi-Fi beacons through a computer program. The Ekahau B4 tags detected these beacons, and measured their signal strengths. The B4 tags were controlled by the Ekahau RTLS Controller (ERC). Information of each detected beacon frame was send to the ERC by the B4 tags. A virtual desktop connection was established with the ERC server in order to both set configurations to the tags and configure ERC to process the received information about detected Wi-Fi beacons. ERC and tags use a protocol named ELP (Ekahau Location Protocol) to communicate with each other. The ELP protocol operates on top of UDP (User Datagram Protocol).

The measurement process was as follows: First, the test person put on the two B4 tags, one in front, one in behind of him. Then he went to a desired measurement location. After this, a script was launched in the ERC, that configured ERC to receive ELP-packets for a time interval of 3 minutes from all the tags that were configured to the ERC. This included also other tags than what where used in this test, but were for some other reason configured to the ERC. The script also set the

Table 3: Example format of RSS-measurement text file

| Place | Attenuation (dB) | TagMac | Time | RSS |
|---|---|---|---|---|
| 11 | 10 | 00:18:8E:22:24:19 | 2013-10-04T17:10:50.399+0300 | -80 |
| 11 | 10 | 00:18:8E:22:24:19 | 2013-10-04T17:11:42.644+0300 | -78 |
| 11 | 10 | 00:18:8E:22:24:19 | 2013-10-04T17:12:20.224+0300 | -77 |
| 11 | 10 | 00:18:8E:22:24:19 | 2013-10-04T16:39:49.444+0300 | -78 |

ERC to dump all the information in the received ELP-packets to a binary dump file. This process was then repeated for all measurement locations in figure 19.

After this followed a preprocessing of the results to a form where they could be analyzed and visualized. First, a program was made that transfered all the binary dump files into a single text file. The code for this program can be found in appendix A. The format of the generated text file can be seen in Table. 3. The first line of the text file represents a header line. The first column indicates the measurement place. The measurement places can be found in Fig. 19. The second column indicates how much attenuation was added to the original signal. Corresponding channel power levels were measured for the attenuations. Those can be seen in Table. 4.

Table 4: The relation of attenuation and measured channel power

| Attenuation (dBm) | Measured Channel Power (dBm) |
|---|---|
| 0 | -9.34 |
| 5 | -14.13 |
| 10 | -18.77 |
| 15 | -23.71 |
| 20 | -29.08 |

Subsequent to the preprocessing of the results of the RSS measurements to a form where they could be analyzed, was time for the actual analysis. From the RSS results, following characteristics were calculated:

1. For each place, for both tags, for each attenuation, the average and variance of the RSS.

2. Plot for attenuation versus RSS for all the RSS measurements.

3. Average difference of the average RSS between the test tags.

R-language was used to make a script which calculated the averages, variances and plotted attenuation versus RSS. The code for the script can be found in appendix B.

Table 5: Results of the RSS measurements with zero attenuation.

| Place | Attenuation | TagMAC | AVG_RSS | VAR_RSS | Count |
|---|---|---|---|---|---|
| 1 | 0 | 00:18:8E:22:24:19 | -62.27 | 0.94 | 51 |
| 1 | 0 | 00:18:8E:22:24:57 | -74 | 0.98 | 41 |
| 2 | 0 | 00:18:8E:22:24:19 | -65.26 | 12.71 | 50 |
| 2 | 0 | 00:18:8E:22:24:57 | -72.82 | 5.51 | 50 |
| 3 | 0 | 00:18:8E:22:24:19 | -75.18 | 4.2 | 38 |
| 3 | 0 | 00:18:8E:22:24:57 | -76.03 | 3.32 | 37 |
| 4 | 0 | 00:18:8E:22:24:19 | -76.29 | 0.56 | 17 |
| 4 | 0 | 00:18:8E:22:24:57 | -75.44 | 0.63 | 52 |
| 5 | 0 | 00:18:8E:22:24:19 | -74.29 | 1.92 | 7 |
| 6 | 0 | 00:18:8E:22:24:57 | -77.67 | 0.89 | 6 |
| 7 | 0 | 00:18:8E:22:24:19 | -75.83 | 0.49 | 52 |
| 7 | 0 | 00:18:8E:22:24:57 | -75.58 | 1.09 | 57 |
| 8 | 0 | 00:18:8E:22:24:19 | -71.98 | 1.46 | 57 |
| 8 | 0 | 00:18:8E:22:24:57 | -75.38 | 1.48 | 8 |
| 9 | 0 | 00:18:8E:22:24:19 | -67.52 | 2.49 | 58 |
| 9 | 0 | 00:18:8E:22:24:57 | -73.73 | 1.54 | 55 |
| 10 | 0 | 00:18:8E:22:24:19 | -65.27 | 1.99 | 59 |
| 10 | 0 | 00:18:8E:22:24:57 | -72.04 | 0.63 | 57 |
| 11 | 0 | 00:18:8E:22:24:19 | -71.39 | 1.11 | 57 |
| 11 | 0 | 00:18:8E:22:24:57 | -78 | 0 | 1 |
| 12 | 0 | 00:18:8E:22:24:19 | -71.35 | 1.63 | 57 |
| 12 | 0 | 00:18:8E:22:24:57 | -77.14 | 3.27 | 21 |
| 13 | 0 | 00:18:8E:22:24:19 | -71.71 | 5.17 | 58 |
| 13 | 0 | 00:18:8E:22:24:57 | -74.07 | 3.32 | 59 |

### 6.1.2 Results and analysis of the RSS measurements

In this subsection, we will review and analyze the results of the RSS measurements that were executed at the Ekahau office. Important thing to note about the results is that the Ekahau B4 tag's threshold for detecting wifi signals was -81 dBm. Therefore in many cases the signal was hardly detected at all, specially with higher attenuation levels of the signal. The full results of the RSS measurements can be found in appendix C. As for closer review, we will now take a look at the results with zero attenuation.

The results for the zero attenuation are shown in Table 5. In Table 5, the meanings of the columns are following: Place means the actual physical location where measurements were made according to Figure 19. Attenuation stands for how much the signal was attenuated by the attenuator in dBm. TagMAC denotes the MAC-address of a B4 tag. AVG_RSS is the average RSS calculated from all the RSS. VAR_RSS is the variance of the RSS measured. Count is the amount of signals detected by the tag. It denotes the amount of signals from which the average and variance is calculated from. It is important to understand, that the lower the

count, then the less signals were detected. The total amount of signals send was 60. The threshold for detecting signal was -81 dBm, as stated earlier. From this we can conclude that the signals that were not detected were in high certainty below the threshold.

From the results we can see that locations near the emulated Wifi AP have a higher signal strength than the ones that are more far away. The highest signal strengths were measured in places 1, 2 and 9. Suprisingly, location 10, despite being more far away from the AP and in another room, has also a very high RSS. A possible reason for this is multipath fading, which can increase the signal strength by summing up different waves positively. Multipath fading is caused by reflection, refraction, scattering and diffraction. Another reason that can partly explain the high RSS is that the wall between rooms A383 and A381 is very thin and consist of a material which does not attenuate the signal much.

The worst RSS was measured in places 3, 4, 5, 6 and 11. Particularly places 5 and 6 are are very bad in terms of the RSS measurements. In those places, only one of the tags has detected signals. That means the other tag was completely unable to detect any signals at all. Though, the tag that did detect signals, detected them only 7 (place 5) and 6 (place 6) times out of 60. From this we can say that most of the signals were undetected in these places, and therefore below the threshold value of -81 dBm. There are multiple reasons why the RSS was low in these places. First is the distance between the tags and the AP. In places 5, 6 and 11 the distance was one of the highest in the set. Second reason is the extra attenuation caused by the walls between the tags and the AP. In places 3, 4, 5 and 6 there were glass walls between the tags and AP. Those glass walls were also covered with metallic venetian blinds, which probably caused much extra attenuation. Between AP location and places 4, 5 and 6 there were two walls the signal needed to penetrate through. First is the wall between A383 and A381, and then there is also the glass wall of room of A381. Of course the direct path between the AP and tag is not the only path the signal will travel, but still, it is a significant factor if the line-of-sight is blocked by two walls.

Let's proceed to review how different attenuations impacted to the measurement results. We were interested in two different attenuation factors. First is how the different output gains of the emulated WiFi AP affected to the measurements. Second is how did the attenuation caused by human body affect to the measurements? As mentioned earlier in the thesis, the test person wore two tags, one in front of him, and one behind. First, we are going to take a look at how the different output gains impacted to the measurements.

In Figure 22, we have a scatter plot of measured RSS with different output gain levels. The higher the attenuation, then the lower the output gain level of the WiFi AP was. As we can see from the Figure 22, the higher the attenuation was, the lower the measured RSS's were and the less signals were detected overall by the tags. Even between attenuations of 0 and 5, there is a clear difference in the amount of signals detected. This can be seen better by looking at the full results of the measurements in Appendix C.

The human body contains approximately 70% of water, and can therefore cause

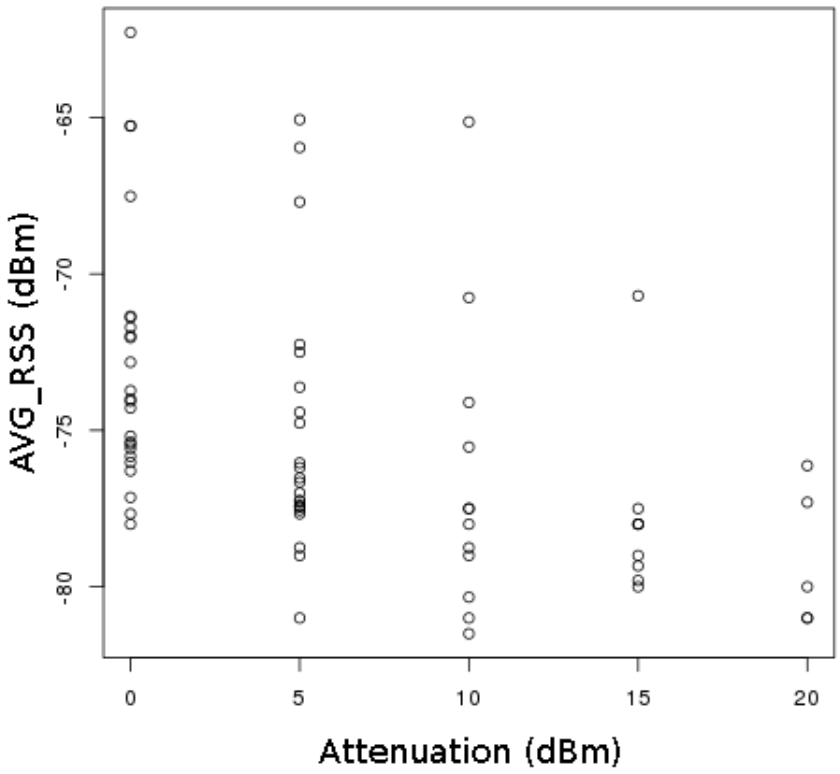


Figure 22: Output gain attenuation vs. received signal strength with all places included.

a significant attenuation to a WiFi signal strength [59]. In the experiment we used 2 different orientations for each measurement place, in order to empirically test how much does human body affect to the RSS. For reference, experiments conducted by Bahl et al. in [44] show that depending on human orientation and location of AP, measured RSS of the AP can vary up to 5 dBm. Let's analyze the affects of human body to our measurements by examining the RSS measurement results with zero attenuation in order to find differences of measured RSS between the two test tags.

As we can see from Table 5, in most of the places, there is a clear difference in the average RSS of the two tags. In places 3 and 4, there was only a minor difference, approximately 1 dBm. In place 10, the difference was almost 7 dBm. One possible reason for the variation of the differences can be the orientation of the user relative to the emulated WiFi AP. In most of the cases, the test user faced towards the AP. Therefore, the tag which was in front of the user, received the direct signal without any human attenuation. And the tag which was behind the user, received the direct signal of the AP after it was first attenuated by the human body. In places of 3 and 4, the direct signal got much attenuation from the glass wall with venetian blinds. It can be that the indirect signals were more dominant in those places. Since user

was facing towards AP and not towards the indirect signals, the human attenuation was similar to both tags.

A script was made to calculate the average difference of the average RSS between the two tags in all places with no extra attenuation used in the output gain. The script can be seen in the end part of Appendix B. Result of the average difference between the tags was 4.84 dBm. The result was pretty similar of the one Bahl et al. got in [44]. Even so, the result is biased in at least two ways. First, the places 5 and 6 were skipped because in those places only one of the test tags had detected signals. Second, because the tags' threshold for detecting a signal was in - 81dBm, the average RSSs' in some cases do not present well the truth about how strong or weak the signal actually was. This can be seen specially in places were many signals were not detected. For example, in place 11, tag 1 detected 57 signals with average RSS of -71,39 dBm. Tag 2 in the same place detected only 1 signal, with RSS of -78 dBm. Because of the threshold, we can conclude that all the other signal RSS were below -81 dBm. Therefore, the value -78dBm does not present the truth about RSS in that place. And therefore, the difference between the tags is even higher in that place.

The main interest of this test was to see whether the signal strength of the emulated WiFi AP was significantly lower in the neighboring rooms of the AP than in the room where the AP was located. If this was the case, then it would give us a good basis for improving the room-level accuracy of a positioning system with a static BLE (Bluetooth Low-Energy) beacon. If we look at the results with zero attenuation, that can be found in Table 5, we can see that in overall, the RSS is higher in the room where AP is located (A383) than in the other rooms (A380 and A381). While this is true in overall, in some cases it does not hold. In places 10 and 9 the RSS is very similar to the RSS of room A383. Also, in places 9 and 13, that are located just behind the wall between A380 and A381, the RSS is is at least close to the RSS in the room of the AP (A380). This is true specially with the tag that is facing towards AP. The main reason for this is probably the wall between A380 and A381, which was narrow and consisted of material that didn't attenuate signals much. Between the rooms A383 and the hallway A380, there is more clear difference in the RSSs. Because of the material and narrowness of the wall between A383 and A381, we can think of it as one of the worst case walls for improving room accuracy with static BLE beacon. But even with it, the overall RSS was different between the rooms. Therefore, we can conlcude, that there is a basis for improving room accuracy with using static BLE beacons. Next, we are going to deploy Ekahau positioning system to Aalto University, and test whether static BLE beacon can improve room-level accuracy, when the walls are much thicker and material is different.

## 6.2 Room-level accuracy test

In this subsection, we will review and analyze the room-level accuracy tests that were done at Aalto University. The Ekahau positioning system was used as a base positioning system, on top of which the tests were made with an Ekahau B4 tag. The

room-level accuracy of the Ekahau positioning system was tested with and without the aid of a static BLE beacon. Plan was to put the BLE beacon into one room, and test how it affects to the room-level accuracy of that room and its neighbours. Before any tests were made, the Ekahau positioning system was deployed to Aalto University, and the integration of the BLE beacon to the positioning system was conducted. The basic methodology of the accuracy test was that we first created a fingerprint model of a selected test area, and then created a test survey with the Ekahau Site Survey program. Then the accuracy of the test survey was tested against the fingerprint model created earlier, both with and without the additional BLE beacon.

### 6.2.1 Deployment of the Ekahau Positioning system

In this subsection the deployment of the Ekahau positioning system to Aalto University will be presented. The deployment of the Ekahau positioning system can be break down to 2 different parts: First is the installation and integration of the different system components. These include Ekahau RTLS Controller (ERC) server program, Ekahau Site Survey (ESS) client program and synchronizing the Ekahau B4 tag to the system. All of the previous components were reviewed in chapter 5. The second part is about surveying a selected test area with the ESS program, in order to create a radio map of fingerprints that is required for positioning.

The ERC is a server program that makes all the positioning estimates based on RSS and fingerprint model it has. A more detailed presentation of the ERC was given in section 5.2. The ERC was installed to Aalto University's virtual server. The virtual server was Windows server 2008 R2 with 64bit version. Remote desktop connection with ssh forwarding was established for managing the ERC server. The installation process of ERC was straightforward: An executable containing the program was run in the server and it was all set to go. A web browser interface to the ERC program can be seen in Figure 15. Before any positioning with the base Ekahau system could be made, two things were required to be done. First, the Ekahau B4 tag had to be connected to the ERC. Second, a fingerprint model of the test area had to be uploaded to the ERC program. Connecting the Ekahau B4 tag to ERC was done using Ekahau Tag Activator program. The Ekahau Tag Activator program comes bundled with the ESS, and its interface can be seen in Figure 23. Before the fingerprint model could be uploaded to the ERC, it had to be first created with the ESS program. Next we will review the process of creating it.

The fingerprint model of the test area was created using the Ekahau Site Survey (ESS) and a B4 tag. Overall description of ESS was given in section 5.1. Let's go through the process of creating the fingerprint model. First, a test area was selected from Aalto University's department of electrical engineering's second floor. The chosen test area is inside the red lined polygon in the Figure 24. The test area figure also shows the location of the Bluetooth Low energy beacon. Then, we chose a tag to be used in the creation of the fingerprint model. Within the ESS program, we selected the earlier connected B4 tag to be used in the survey. The operating principle of using the ESS with a B4 tag is following: The B4 tag will do all the

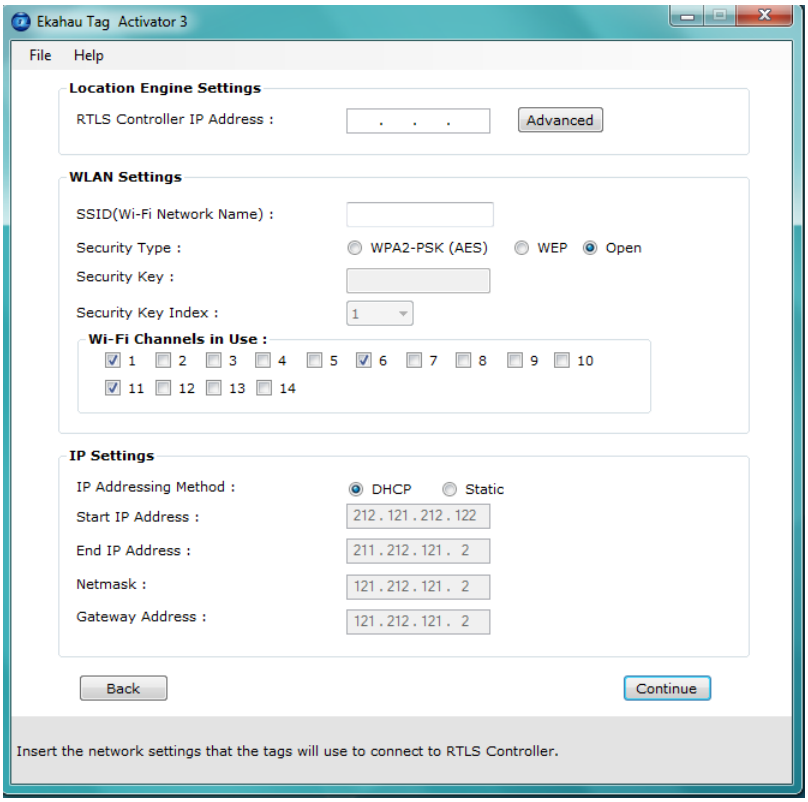


Figure 23: The Ekahau Tag Activator 3

WiFi scanning, send results of them to ERC, which will relate those results to the ESS program, which will then make a model based on them. Hence, the B4 tag is not directly connected with the ESS program, but through the ERC controller server program. The interface of selecting a tag to be used in the survey can be seen in Figure 25.

Surveying an area in ESS is done by using a war-walking method, described in more detail in sections 3.5.1 and 5.1. Surveying the test area included walking with a laptop running the ESS program, and pinpointing user's physical location on a map within the ESS, while simultaneusly wearing the B4 tag which was continuously scanning for WiFi networks.

After the basic surveying of the test area was done, few enhancements were made to the model in the ESS program for achieving better positioning. First, we determined a pixel to meter ratio of the map. A pixel to meter ratio allowes ERC to take into account physical distances in the positioning algorithm. Second, we draw rails according to the physical constraints of the test area as described in the last part of the section 5.1. Rails indicate how trackable objects can move. This enhances the accuracy of the positioning system, because it then knows much more about possible movement, and can filter away certain hypotheses of the movement. A good example of filterable hypotheses is movement through walls. These measurement rails for the fingerprint model in the test area can be seen in Figure 26. As can be seen from the Figure, the rails cover more area than the actual test area. This is because we did

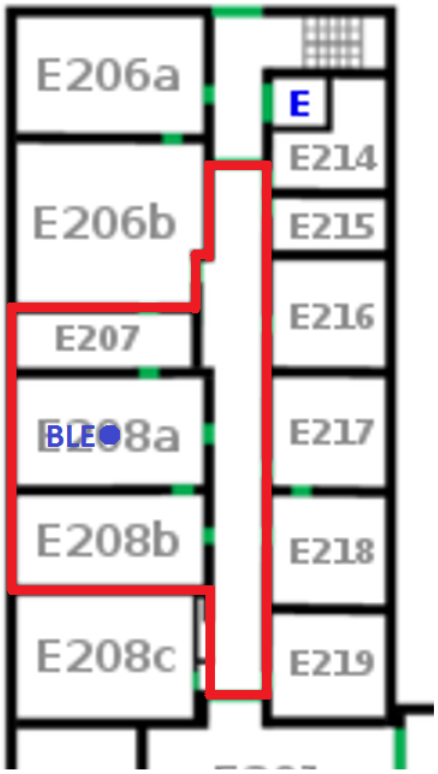


Figure 24: Test area of the room accuracy tests and the position of the BLE beacon

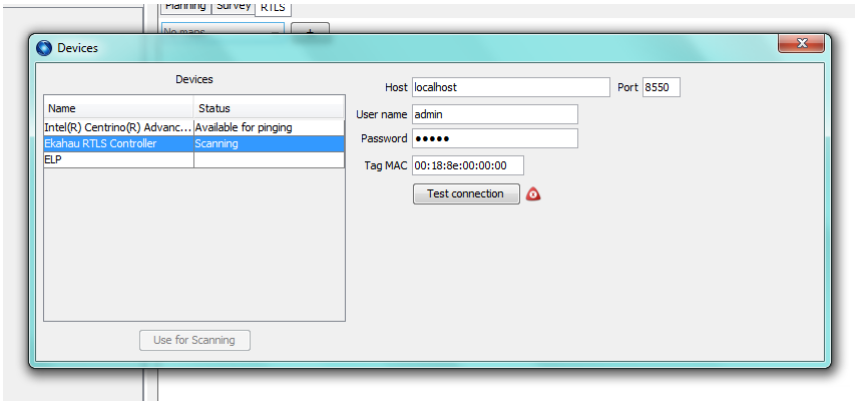


Figure 25: Tag selection interface in the Ekahau Site Survey

model a bigger area in the same survey.

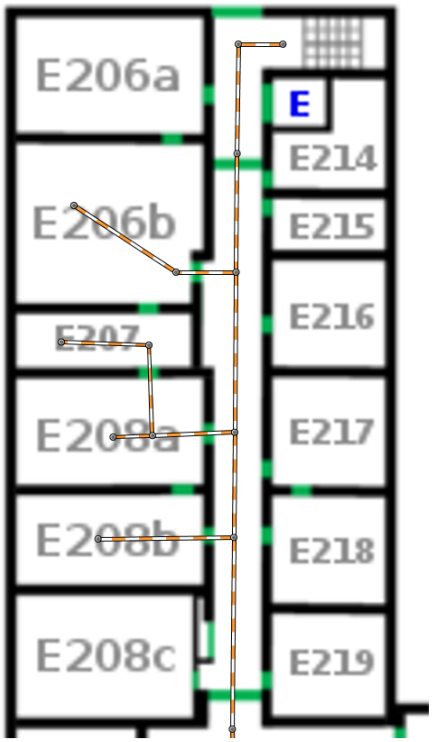


Figure 26: Rails in the test area

The last improvement made to the fingerprint model was drawing of zones. Within the ESS program, user can separate different areas to zones. Drawing logical areas into different zones will enhance the positioning quality. In more detail, the operation of the positioning algorithm's particle filter is improved by determining zones. Determining zones for each room and hallway enables easy reviewing of room-accuracy. The drawed zones for the test area can be seen in Figure 27.

With all the aforementioned improvements, the fingerprint model of the test area was ready to be submitted to ERC. The ESS program creates the model as a packed file with the extension of .esx. Inside this model file are all the details of the model: surveys, rails, zones, detected APs, tags, map images, pixel to meter ratios, and most importantly, the received signal strengths with their location data. The fingerprint model was then submitted to the ERC through ERC's web interface, which can be seen in Figure 15. This completed the deployment of the Ekahau positioning system, which was now ready for tracking all tags that were connected to it in the test area. Next, we will review the integration of the BLE beacon with the Ekahau positioning system.

### 6.2.2 Integration of a BLE beacon to the system

In this subsection, the integration of a BLE beacon to the Ekahau positioning system will be presented. We will present the Bluetooth low energy beacon used in the test,

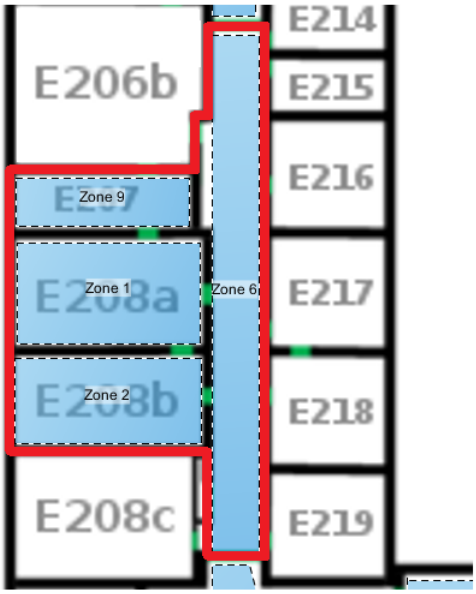


Figure 27: The zones of the test area.

and its integration to the positioning system. The main goal of the thesis was to invent a method for improving room accuracy by using a static low energy bluetooth beacon. In order to achieve this, the system needs to recognize the beacon somehow. A decision was made that the BLE beacon would be integrated to the system as if it would be a regular WiFi AP found during a survey. Then the positioning algorithm would take it into account as a regular WiFi AP.

The BLE beacon used in the tests was a tiny coin-cell battery based smart beacon, seen in Figures 28, 29. Once a coin battery is inserted to the smart beacon, it starts sending BLE beacon frames in continuous interval.

Figure 28: The Bluetooth Low-Energy beacon used in the room-level accuracy tests

The main question was, that how could the system detect the Bluetooth low energy beacon? Because the B4 tags were unable to detect bluetooth low-energy beacons, another method had to be invented as how to detect and measure the signals of the BLE beacon. The solution found was to use an android phone for

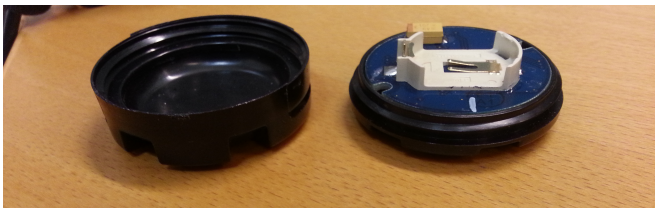

Figure 29: The Bluetooth Low-Energy beacon used in the room-level accuracy tests

scanning the bluetooth low energy beacon, and then integrate those scanning results with the tag's scanning results. Assuming BLE capability was added to the B4 tags in the future, this would then simulate the real situation, when the tag would be byself able to detect BLE beacon signals.

In order for the positioning system to recognize the BLE beacon, the signals of the BLE beacon have to be integrated in the fingerprint model created earlier, and also to forthcoming test surveys, which will test the room-level accuracy against the model. Therefore, the plan was to scan the BLE beacon signals with an android device simultaneously, while making the surveys with the B4 tag. Then, after the surveys were made, the BLE scan results from android device were added to corresponding surveys in the ESS-model.

An Android software capable of scanning and measuring BLE signals was developed, and labelled as BLE scanner. The source code for this software can be found in Appendix D. The BLE scanner performs BLE scanning by accessing the core BLE sensors of an Android smartphone by using Google API. It stores scanning records to a flat file (BLE_ScanResult.txt), inside the device's memory. The scan records contain three attributes:

1. The MAC address of the detected BLE beacon
2. The received signal strength of the detected BLE beacon
3. The timestamp when the BLE beacon's signal was detected and measured

These aforementioned scan records can then be integrated into the fingerprint model created with the ESS-program. Let's go through the process of integration we made between BLE scan records and an ESS-model. An ESS-model is a compressed file with an .esx extension. Inside the compressed file, one can find everything related to a model. These include the map images that were used, general properties of the model, and separate files for each survey. For each survey, there are separate files for the survey's overview (XML), and for the survey's RSSI measurements (binary). In the file that describes the project overview (XML), all APs found during all the surveys of the model are listed. Hence, the BLE beacon has to be added there as well. Also, in the overview file of a specific survey, all APs found during the survey are listed, and therefore the BLE beacon has to be added there as well, if it was detected during the survey. The last part of the integration is to add the BLE RSS-data into a corresponding survey binary file. For this purpose, a Java-based script was developed, and can be seen in Appendix E. After these operations are done, the model can be compressed back to a .esx model file, and the integration is complete.

### 6.2.3 Test scenario and test survey

In this subsection, the room-level accuracy test scenario is presented in more detail. As stated earlier in section 6.2, the basic methodology of the room-level accuracy test was as follows:

1. Fingerprint model of the test area was created with the ESS-program and the B4 tag
2. Test surveys were created with the ESS-program and the B4 tag
3. The room-level accuracy of the test surveys was tested against the fingerprint model, which was created earlier

Aforementioned steps were made with the integration of a BLE beacon and the positioning system, described in previous subsection. Hence, while creating the Fingerprint model with the ESS-program, we simultaneously scanned BLE signals with Android software. The BLE scanning results were then added to the fingerprint model. This same procedure was done when the test surveys were made with the ESS program. Before we present the test surveys and how their room-level accuracy was tested, let's review the integrated fingerprint model, in order to understand the local Wi-Fi environment of the test area.

The test area was in Aalto University's School of electrical engineering (SEE) second floor E-wing, as seen in Figure 24. The fingerprint model of the test area was created by using the ESS-program. Within the ESS program, the test area's Wi-Fi RSS can be visualized. Also, information about detected APs can be observed through ESS. The visualization of WiFi RSS is shown in Figure 30. In the Figure 30, the visualization of the WiFi RSS is based on the highest RSS when all detected APs are taken into account. The visualization is based on real RSS, but of course, it is an estimate. The real scannings come from certain points on the map. Based on those points, the ESS program can estimate and visualize the RSS in the area.

The RSS in the fingerprint model is visualized in the ESS-program by different colours from green to red. The more green the colour is, the higher the RSS is. The more red the colour is, the more low the RSS is. In the Figure 30, the places of closest WiFi APs and the place of the BLE beacon is shown. From the ESS program model we can observe that during the calibration surveys, a total of 20 different WiFi APs were detected. Because the calibrated area was larger than the test area, not all of the APs were necessary detected in the test area. Only handfull of APs had a good RSS in the test area. This certainly limits the accuracy of the positioning system. Next, let's take a look at the BLE beacon's RSS in the fingerprint model, seen in Figure 31.

As seen in Figure 31, the RSS of the BLE beacon in the fingerprint is high only in the room where it is located (E208a). In it's neighboring rooms, the signal was detected, but the RSS was low.

It is time to review how the test surveys were made, and how their room-level accuracy was tested against the fingerprint model. As stated earlier, the room-level accuracy was tested with and without the BLE beacon, in order to see whether

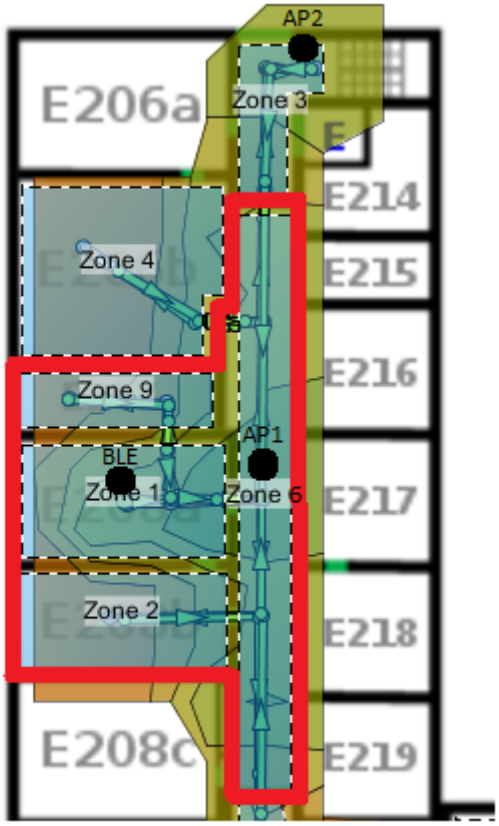


Figure 30: The Wifi RSS in the test area's fingerprint model

there is difference in the accuracy. The creation of test surveys was similar to the creation of a fingerprint model. It was done by creating a new survey with ESS, and marking it as a test survey. For the sake of clarity and further processing, a completely new model was created for the test survey. In practise, this means we had two model files with an .esx extensions, one for the fingerprint model, and one for the test survey. But, because we wanted to test the accuracy with and without the BLE beacon, we created total of 4 models: Fingerprint models with and without BLE, and test models with and without BLE.

The creation of the test survey was similar to the creation of a fingerprint model survey. We first configured the B4 tag to make all the scannings in ESS for the survey, then started walking in the test area with a laptop running the ESS and wearing the B4 tag. On top of this, as we started the survey, we also started the android application called BLE scanner, in order to capture the BLE beacon signals. Then we walked along the test area for 15 minutes. During the survey, we marked our position to the survey model. Because the correct positions are marked in the test survey, the accuracy of the test survey can be easily calculated. So the scan-data of the survey is used as an input to the positioning algorithm, and then the positions calculated by the algorithm are compared with the correct positions of the test survey. During the test survey, total of 960 scans were made. Duration of one

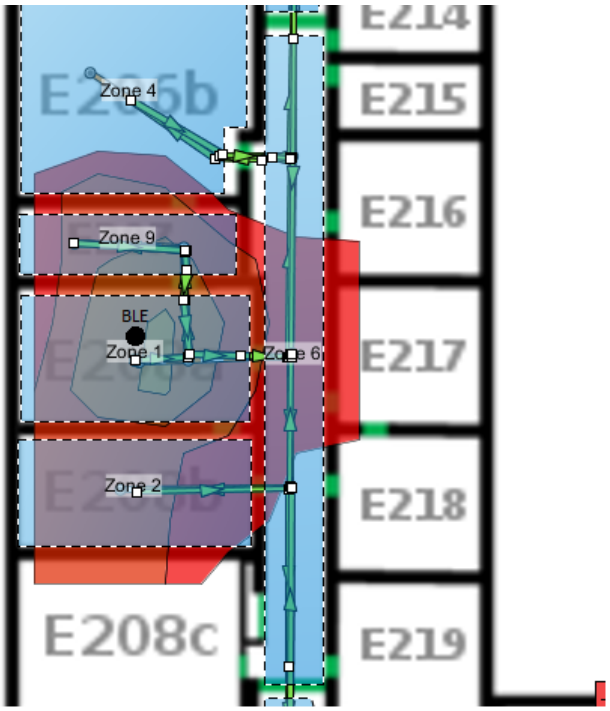


Figure 31: The BLE RSS in the test area's fingerprint model

scan is approximately 100ms.

Once the test survey was made, it was time for the room-level accuracy analysis. The room-level accuracy analysis was done by using a program called RTLSToolbox. RTLSToolbox (Figure 32) can run the Ekahau positioning algorithm with test data, and provide various accuracy statistics. All it requires is the fingerprint model and a test survey as an input. The fingerprint model was set to the program as a model file with the .esx extension. For the test survey, a modification for the file type was required. The RTLSToolbox required a binary dump file for the test survey. Therefore, the model, which consisted the test survey, was converted to a dump file by using a ready made program called EsxSurveyAsSignalDump. Then the fingerprint model and the test survey binary dump file was set to the RTLSToolbox as an input, and the accuracy analysis was executed. Behind the scenes, RTLSToolbox runs the Ekahau positioning algorithm with the test survey, and outputs many accuracy related results into a project folder to be reviewed. Next, let's review and analyze the results of the accuracy tests.

### 6.2.4 Results and analysis of the room-level accuracy test

In this subsection, the results of the room-level accuracy test are reviewed and analyzed. The room-level accuracy tests were made with the RTLSToolbox program, as described in the previous subsection. The tests were run with and without the integration of BLE beacon data. In practise, separate fingerprint model files and test survey model files and dumps were made for the two cases. Then the accuracy

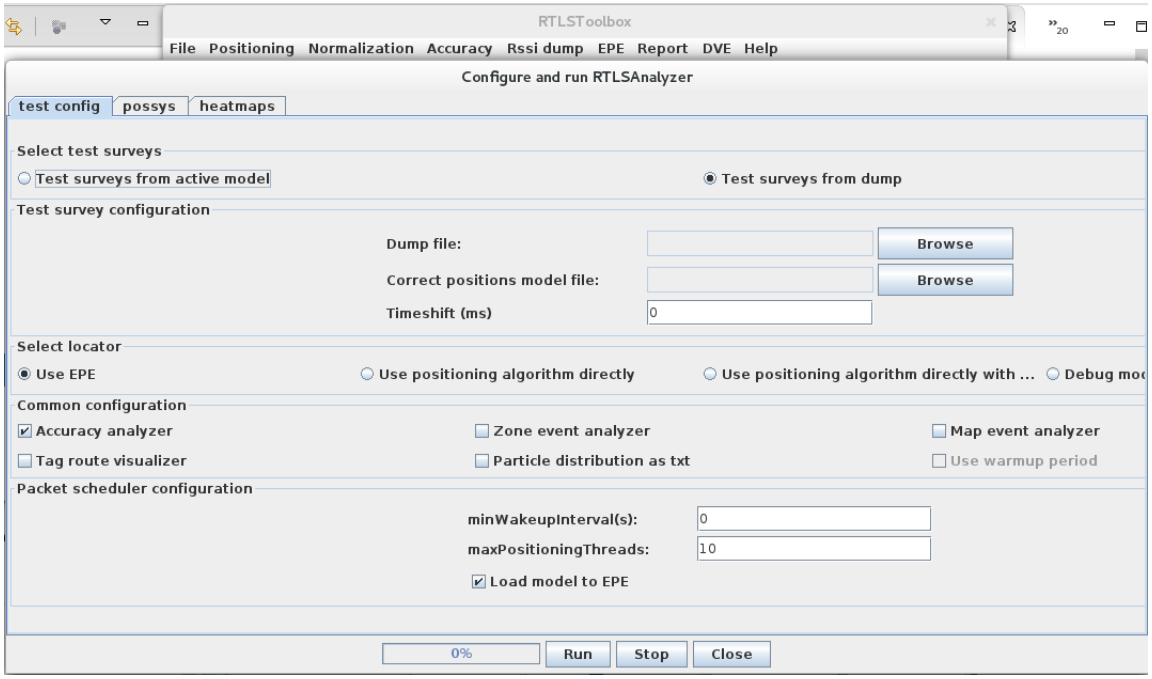


Figure 32: RTLSToolbox's RTLSAnalyzer

Table 6: The room-level accuracy test results

| Room | Without BLE % | With BLE % |
|---|---|---|
| E208a (zone 1) | 27,76 | 46,53 |
| E208b (zone 2) | 48,69 | 78,65 |
| E207 (zone 9) | 13,47 | 97,83 |
| Hallway (zone 6) | 65,66 | 46,46 |

of the test surveys were tested separately with the RTLSToolbox program.

Let's review the results of the room-level accuracy tests. The test area was divided into zones in a way that each zone represented one room, as seen in Figure 27. For these zones, the RTLSToolbox was able to calculate accuracy statistics. The accuracy was tested with and without the BLE beacon data, and the room-level accuracy results can be seen in Table. 6. The same results, visualized in the map context can be seen in Figure 33. What the results tell us is that how accurately the positioning algorithm was able to locate the test person into a specific room. Which means that, during the time the test person was in a specific room, how many of the positioning estimates were actually estimated to that room? As we can see from the results, for almost every room, the room-level accuracy is higher when BLE is used. The total room-level accuracy without BLE was 34,24%, and with BLE it was 64,81%. Hence, the total room-level accuracy improvement gain from the use of BLE beacon was 30,57 percentage points.

Before doing more analysis, it is important to see from how many positioning estimates the accuracy results derived from. The amount of estimations can be


Figure 33: The results of the room-level accuracy test

Table 7: The amount of positioning estimates

| Room | Without BLE | With BLE |
|---|---|---|
| E208a (zone 1) | 389 | 389 |
| E208b (zone 2) | 267 | 267 |
| E207 (zone 9) | 193 | 184 |
| Hallway (zone 6) | 99 | 99 |

seen in Table 7. The estimates are based on the correct positions. It means how many estimates were made when the person really was in that room. Therefore, the amount of estimates between the cases when BLE was used and not used are almost the same. During the test survey, the test person spend majority of his time in the room where BLE beacon was located, room E208a. It was the biggest room, and of course the room which we were mostly interested in. But, as can be seen from the results, the room-level accuracy is increased not only in that room, but also almost in all other rooms as well. In fact, the biggest improvement in the room-level accuracy was achieved in room E207.

The room accuracy increased in all the other rooms except in the Hallway. In the Hallway, the room-level accuracy was decreased 19 percentage points when BLE was used. There are many causes which may have affected to the accuracy. First, the time spent in the hallway during the test survey was low compared to the rooms. Second, in the fingerprint model, the whole hallway was surveyed, but in the test survey, only a small part of the hallway was surveyed. Third, the area of the hallway that was covered during the test survey was directly below a WiFi AP. Fourth, the

Table 8: Positioning accuracy in meters

| Room | Without BLE (m) | With BLE (m) |
|---|---|---|
| E208a (zone 1) | 3,85 | 2,15 |
| E208b (zone 2) | 9,34 | 3,55 |
| E207 (zone 9) | 3,77 | 0,91 |
| Hallway (zone 6) | 8,33 | 5,39 |

hallway area in the test survey was in the border of the BLE's coverage area, which means that the detection rate for the BLE signal was weak.

Let's further analyze the results of the room-level accuracy test. One major attribute that affected to the results was the material and thickness of the walls at Aalto University's SEE. The walls were thick and made of bricks. This caused enough attenuation for the BLE signal's RSS to be clearly lower in the neighboring rooms. Which made the fingerprint more different in the rooms, and lead to a more accurate positioning. The addition of the BLE alone made the fingerprints more unique, even without the aid of the walls. But the walls gave extra attenuation which improved the performance.

In addition to room-level accuracy, the RTLSToolbox was able to calculate accuracy in terms of meters as well. The positioning accuracy in meters with and without the BLE beacon is shown in Table 8. Without BLE, the overall positioning accuracy was 5,87m. With the aid of the BLE beacon, the overall positioning accuracy was 2,7m. From these results we can conclude that the integration of the BLE beacon to the current scenario did not only increase the room-level accuracy, but affected clearly also to the overall accuracy.

Why did the room-level accuracy increase when BLE beacon was integrated with the positioning system? The main reason is, that the addition of the BLE beacon made the test area's WiFi fingerprints more unique. The reason why the increasement was quite substantial is that the amount of WiFi APs in the test area was not optimal for the positioning system. Therefore the room-level accuracy without the BLE isn't very good. And, even with BLE, there was certainly room for improvement in the room-level accuracy. Second reason why the BLE increased the accuracy as much is the walls of the test areas rooms. The walls were thick, and made of bricks. This certainly increased the uniqueness of the fingerprints.

# 7 Summary and future work

This thesis has presented a method for increasing the room-level accuracy of indoor positioning by using a static low-cost beacon. A bluetooth low energy module was used in the room-level accuracy test conducted in this thesis. Two different kind of tests were executed in the thesis. First test was an RSS measurements -test, where the RSS of an emulated static WiFi AP was measured in different locations indoors, in order to study indoor behaviour of the WiFi signal. The RSS measurement test's goal was to find whether it would be possible to improve the room-level accuracy by using a low-cost static beacon. In the second test, the room-level accuracy of the Ekahau positioning system was tested with and without the aid of an BLE beacon.

A literature research was conducted first in the thesis about indoor positioning, related technologies and indoor propagation. First, the base technologies used in indoor positioning were covered. Then, different indoor positioning approaches were reviewed. Next, the mechanisms of indoor propagation were presented. And last, the Ekahau positioning system was overviewed.

Let's go through the main results of the thesis. As stated earlier, two different kind of tests were made in the thesis, the RSS measurement test and the room-level accuracy test. The RSS measurements test was made in the Ekahau office. In the test, the RSS of a simulated WiFi AP was measured in different locations between two rooms and a corridor. The simulated WiFi AP was put into one room, and then the RSS was measured in that room and also in the neighboring rooms. The test showed, that the RSS was clearly lower in the neighboring room in overall. Though in a few spots just behind the wall, the RSS was close to the RSS measured in the room of the AP. Partial reason for this was that the wall between the rooms was thin, and consisted of material which lead only to low attenuation.

There are multiple ways the RSS measurement test could have been improved. Now the test was made only in the Ekahau office with one AP in one room. We could have made measurements with two or more beacons simultaneusly, and see how that would have affected to results. The threshold of signal detection of the Ekahau B4 tags could have been rised in order to get more knowledge of signal propagation. The measurements could have been done in multiple rooms, and with multiple different wall materials. Still, the test results gave us enough evidence, that the proposed method of static beacons could improve room-level accuracy. This is because the signal strength was clearly lower in the neighboring room, which was a requirement for the proposed method.

The room-level accuracy test was done at Aalto University school of electrical engineering. The test included deployment of the Ekahau positioning system, integration of Bluetooth low-energy beacon to the positioning system, creation of test scenario, running room-level accuracy tests and analysis of them. The Bluetooth low-energy beacon was integrated with the Ekahau positioning system in a way that the system regarded it as a normal WiFi AP. Bluetooth low-energy Android program was made and used in the test in order to capture Bluetooth beacons. According to the test results, the overall room-level accuracy increased from 34% to 65% when BLE beacon was used. The regular positioning also increased, from 5,87m to 2,70m.

The room-level accuracy test could have been improved by choosing a larger test area, testing with multiple beacons, testing accuracy betweeen different floors and increasing the amount of positioning estimates (increase in scenario time). Also, the proposed method could have been improved by taking the BLE specific attributes into account. Because now it was regarded just as a normal WiFi AP. The BLE beacon interval and signal power could have been taken into account in the positioning algorithm. One interesting idea for improving the room-level accuracy is use of ultrasound beacons instead of BLE beacons. Because ultrasound waves gets attenuated much more by the walls of rooms than BLE beacon waves, the fingerprints between rooms would be more unique when ultrasound is used. Ultrasound waves could be used as a binary indicator, for whether or not a blind node is inside a room. [39]

The room-level accuracy test showed, that at least under certain conditions, the room-level accuracy of a positioning system can be improved by applying static beacons. The wall material and thickness are essential factors in the attenuation of beacon signals, and therefore also essential in effectiveness of the proposed method.

# References

[1] Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[2] Sampsa Lindroos. Wireless local area network in a residential building. Master's thesis, Aalto University, 2009.

[3] Newton H. and Schoen S. *Newton's telecom dictionary*. Flatiron Publishing, 2007.

[4] Sriharsha Kuchimanchi. Bluetooth low energy based ticketing systems. Master's thesis, Aalto University, School of electrical engineering, 3 2015.

[5] S.S. Chawathe. Beacon placement for indoor localization using bluetooth. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 980–985, Oct 2008.

[6] Finkenzeller K. *RFID-Handbuch*. John Wiley & Sons, 1999.

[7] L.M. Ni, Yunhao Liu, Yiu Cho Lau, and A.P. Patil. Landmarc: indoor location sensing using active rfid. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 407–415, 2003.

[8] Donggeon Lee, Seongyun Kim, Howon Kim, and Namje Park. Mobile platform for networked rfid applications. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 625–630, April 2010.

[9] Small rfid chip compared to a grain of rice. `http://en.wikipedia.org/wiki/Radio-frequency_identification`. Accessed: 2015-05-14.

[10] S.S. Saad and Z.S. Nakad. A standalone rfid indoor positioning system using passive tags. *Industrial Electronics, IEEE Transactions on*, 58(5):1961–1970, 2011.

[11] S.P. Subramanian, J. Sommer, S. Schmitt, and W. Rosenstiel. Ril - reliable rfid based indoor localization for pedestrians. In *Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference on*, pages 218–222, 2008.

[12] Qing Fu and Guenther Retscher. Active rfid trilateration and location fingerprinting based on rssi for pedestrian navigation. *The Journal of Navigation*, 62:323–340, 4 2009.

[13] S. Gezici, Zhi Tian, G.B. Giannakis, Hisashi Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70–84, 2005.

[14] James D. Taylor. *Introduction to Ultra-Wideband Radar SSystem.* CRC Press, 1995.

[15] M.G.M. Hussain. Ultra-wideband impulse radar-an overview of the principles. *Aerospace and Electronic Systems Magazine, IEEE*, 13(9):9–14, 1998.

[16] Yanying Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys Tutorials, IEEE*, 11(1):13–32, First 2009.

[17] D.J. Murphy. Characteristics of a small low cost inertial measurement unit. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*, pages 75–87, Aug 1998.

[18] Hannes Hyvönen. Thermomechanical and mechanical characterization of a 3-axial mems gyroscope. Master's thesis, Aalto University, 2011.

[19] Gyroscope. `http://en.wikipedia.org/wiki/Gyroscope`. Accessed: 2015-05-14.

[20] Acar C. and Shkel A. *MEMS Vibratory Gyroscopes: Structural Approaches to Improve Robustness.* Springer-Verlag, 2008.

[21] Mika Kämäräinen. *Low-power front-ends for capacitive three-axis accelerometers.* PhD thesis, Aalto University, 2010.

[22] Pyry Peitso. Space weather instruments and measurement platforms. Master's thesis, Aalto University, 2013.

[23] Wonho Kang, Seongho Nam, Youngnam Han, and Sookjin Lee. Improved heading estimation for smartphone-based indoor positioning systems. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 2449–2453, Sept 2012.

[24] Binghao Li, B. Harvey, and T. Gallagher. Using barometers to determine the height for indoor positioning. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–7, Oct 2013.

[25] Zheng Yang and Yunhao Liu. Quality of trilateration: Confidence based iterative localization. In *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, pages 446–453, June 2008.

[26] Apostolia Papapostolou and Hakima Chaouchi. Integrating rfid and wlan for indoor positioning and ip movement detection. *Wirel. Netw.*, 18(7):861–879, October 2012.

[27] M. Vossiek, L. Wiebking, P. Gulden, J. Weighardt, and C. Hoffmann. Wireless local positioning - concepts, solutions, applications. In *Radio and Wireless Conference, 2003. RAWCON '03. Proceedings*, pages 219–224, Aug 2003.

[28] Ferit Ozan Akgul and K. Pahlavan. Aoa assisted nlos error mitigation for toa-based indoor positioning systems. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–5, Oct 2007.

[29] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *Signal Processing Magazine, IEEE*, 22(4):41–53, July 2005.

[30] N.S. Correal, S. Kyperountas, Q. Shi, and M. Welborn. An uwb relative location system. In *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*, pages 394–397, Nov 2003.

[31] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, Nov 2007.

[32] Simon Saunders and Alejandro Savagón-Zavala. *Antennas and Propagation for Wireless Communication Systems*. WILEY, 2007.

[33] H. Ikonen J. Niittylahti J. Latvala, J. Syrjärinne, editor. *Evaluation of RSSI-based Human Tracking*, volume 4 of *Proceedings of the 2000 European Signal Processing Conference*, Tampere, Finland, 2000.

[34] C. Beder and M. Klepal. Fingerprinting based localisation revisited: A rigorous approach for comparing rssi measurements coping with missed access points and differing antenna attenuations. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–7, Nov 2012.

[35] Sang-Hyun Bae Mi-R im Song, Jin-Young Moon. Efficient indoor positioning by hybrid algorithm. In *Proceedings, The 3rd International Conference on Circuits, Control, Communication, Electricity, Electronics, Energy, System, Signal and Simulation*, volume 25, 2013.

[36] Seong-Eun Kim, Yong Kim, Jihyun Yoon, and Eung Sun Kim. Indoor positioning system using geomagnetic anomalies for smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–5, Nov 2012.

[37] J. Rantakokko, J. Rydell, P. Stromback, P. Handel, J. Callmer, D. Tornqvist, F. Gustafsson, M. Jobs, and M. Gruden. Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *Wireless Communications, IEEE*, 18(2):10–18, April 2011.

[38] F. Keller, T. Willemsen, and H. Sternberg. Calibration of smartphones for the use in indoor navigation. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8, Nov 2012.

[39] S. Holm. Ultrasound positioning based on time-of-flight and signal strength. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–6, Nov 2012.

[40] U. Birkel and M. Weber. Indoor localization with umts compared to wlan. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–6, Nov 2012.

[41] Teemu Roos, Petri MyllymÃ¤ki, Henry Tirri, Pauli Misikangas, and Juha SievÃ¤nen. A probabilistic approach to wlan user location estimation. *International Journal of Wireless Information Networks*, 9(3):155–164, 2002.

[42] A.W.T. Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization. *Mobile Computing, IEEE Transactions on*, 9(11):1551–1562, Nov 2010.

[43] Lei Xie, Yuhao Wang, and Xia Xue. A new indoor localization method based on inversion propagation model. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4, Sept 2010.

[44] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784 vol.2, 2000.

[45] P. Myllymäki, P. Kontkanen, T. Roos, K. Valtonen, J. Lahtinen, H. Wettig, A. Tuominen, and H. Tirri. Error estimate concerning a target device's location operable to move in a wireless environment, June 16 2005. US Patent App. 10/999,220.

[46] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *In Proceedings of International Conference on Ubiquitous Computing (UbiComp*, pages 88–106, 2004.

[47] Philipp Vorst, Juergen Sommer, Christian Hoene, Patrick Schneider, Christian Weiss, Timo Schairer, Wolfgang Rosenstiel, Andreas Zell, and Georg Carle. Indoor positioning via three different rf technologies. In *RFID Systems and Technologies (RFID SysTech), 2008 4th European Workshop on*, pages 1–10, June 2008.

[48] D. Fox. Kld-sampling: Adaptive particle filters. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[49] Wei Leong Khong, Wei Yeang Kow, Yit Kwong Chin, Mei Yeen Choong, and K.T.K. Teo. Enhancement of particle filter resampling in vehicle tracking via genetic algorithm. In *Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on*, pages 243–248, Nov 2012.

[50] Hui Wang, H. Lenz, A. Szabo, J. Bamberger, and U.D. Hanebeck. Wlan-based pedestrian tracking using particle filters and low-cost mems sensors. In *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on*, pages 1–7, March 2007.

[51] Jeanette Mulligan. A performance analysis of a csma multihop packet radio network. Master's thesis, VirginiaTech, Electrical Engineering, 1997.

[52] C. A. Balanis. *Advanced engineering electromagnetics*. John Wiley & Sons, New York, 1989. ISBN 0-471-62194-3.

[53] J. M. Keenan and A. J. Motley. Radio coverage in buildings. *BT Technology Journal*, pages 19–24, 1990.

[54] B-S Lee, A.R. Nix, and J.P. McGeehan. Indoor space-time propagation modelling using a ray launching technique. In *Antennas and Propagation, 2001. Eleventh International Conference on (IEE Conf. Publ. No. 480)*, volume 1, pages 279–283 vol.1, 2001.

[55] Ekahau. Ekahau 2013 product guide. `http://www.ekahau.com/userData/ekahau/documents/datasheets/2013_Product_Guide_sept-13.pdf`, 9 2013.

[56] Ekahau. Wi-fi tag for tracking assets. `http://www.ekahau.com/userData/ekahau/documents/datasheets/A4_datasheet_letter.pdf`, 2013.

[57] Ekahau. B4 wi-fi badge tag. `http://www.ekahau.com/userData/ekahau/documents/datasheets/B4_datasheet_letter.pdf`, 2013.

[58] Ekahau. W4 wearable wi-fi tag. `http://www.ekahau.com/userData/ekahau/documents/datasheets/W4_Datasheet_letter.pdf`, 2013.

[59] Francescantonio Della Rosa, Mauro Pelosi, and Jari Nurmi. Human-induced effects on rss ranging measurements for cooperative positioning. *International Journal of Navigation and Observation*, 2012(959140):13, 2012.

# Appendices

## A DumpToText Java program

```
/**
 * This content is released under the (http://opensource.org/licenses/MIT)
 * MIT License.
 *
 * Copyright (c) 2015 Ekahau, Ville Miekk-oja.
 */

package com.ekahau.positioning.dumptotext;

import java.io.BufferedWriter; import java.io.File;
import java.io.FileNotFoundException; import java.io.FileWriter;
import java.io.IOException; import java.text.SimpleDateFormat;
import java.util.ArrayList; import java.util.Collections;
import java.util.Date; import java.util.HashSet;
import java.util.Set; import java.util.List;
import com.ekahau.epelibs.common.Mac;
import com.ekahau.epelibs.rssi.RssiStorageEntry;
import com.ekahau.rtlslibs.dump.DumpReader;
import com.ekahau.rtlslibs.dump.DumpReaderFactory;
import com.ekahau.rtlslibs.dump.RssiEntryIterator;

public class DumpToText {

  private static final SimpleDateFormat SDF = new SimpleDateFormat(
    "yyyy-MM-dd'T'HH:mm:ss.SSSZ");
  public static int isTrue;
  // set your directory path here
  private static String myDirectoryPath = "tests";
  private static final void usage() {
  System.err.println(
    "Usage: java DumpPrint rssi|hypo|req|obsprob filename working_dir");
  }

  //Pretty print rssi entry
  public static final String rssiEntryToString(RssiStorageEntry pEntry) {
    StringBuilder ret = new StringBuilder();
    String tagMac = Mac.toString(pEntry.getMac());
    if (tagMac.equalsIgnoreCase("00:18:8E:22:24:57") || tagMac.equalsIgnoreCase("
        00:18:8E:22:24:19")) {
      isTrue = 0;
      for (RssiStorageEntry.ApChunk ae: pEntry.getRssis()) {
        String apMac = Mac.toString(ae.getApMac());
        apMac = apMac.substring(0, apMac.length() - 1);
        if (apMac.equalsIgnoreCase("00:18:8e:77:1c:9d")) {
          ret.append(tagMac);
          ret.append(" " + SDF.format(new Date(pEntry.getTimeStamp())) + " " );
          ret.append(ae.getRssiOriginal());
          ret.append("\n");
          isTrue = 1;
        }
      }
      if (isTrue == 1) {
        //The tag heard the test AP
        return ret.toString();
      }
      else {
        //The tag didn't hear the test AP
        return tagMac;
      }
    }
    else {
```

```
    //the rssiEntry was not for one of the test tags
    return "";
  }
}

public static void main(String[] args) throws IOException {
  if (args.length != 2) {
    usage();
    System.exit(1);
  }
  String action = args[0];
  String fileNamePrefix = args[1];
  DumpReader rdr = null;
  try {
    File file = new File("Antenni_sijainti_A.txt");
    BufferedWriter output = new BufferedWriter(new FileWriter(file));
    output.write("Place␣Attenuation␣TagMac␣Time␣RSS\n");
    File dir = new File(myDirectoryPath);
    BufferedWriter sumsWriter = new BufferedWriter(
      new FileWriter("Sums.txt"));
    sumsWriter.write(
      "Place␣Attenuation␣Totalsize␣missed_24:57␣missed_24:19\n");
    File[] folderList = dir.listFiles();
    for (File child : folderList) {
      String folderName = child.getName();
      //parse attenuation and mes.location from folder name
      String[] parts = folderName.split("[vV]");
      String locationUnmodified = parts[0];
      String[] locationParts = locationUnmodified.split("␣");
      String location = locationParts[1];
      String attenuationUnmodified = "v" + parts[1];
      String[] attenuationParts = attenuationUnmodified.split("␣");
      String attenuation = attenuationParts[1];
      if (child.isDirectory()) {
        File[] fileList = child.listFiles();
        for (File child_2 : fileList) {
          String filename = child_2.getName();
          if (filename.equals(fileNamePrefix)) {
            try {
              rdr = DumpReaderFactory.singleFileReader(action, myDirectoryPath +"
                  /"+folderName + "/" + fileNamePrefix);
            } catch (FileNotFoundException ex) {
              // Not found
            }
            if (rdr == null) {
              // try prefix
              rdr = DumpReaderFactory.multipleFileReader(fileNamePrefix);
            }
            RssiEntryIterator it = rdr.rssiIterator();
            Set<Long> timestamps = new HashSet<Long>();
            List<String> tags = new ArrayList<String>();
            while (it.hasNext()) {
              RssiStorageEntry r = it.next();
              String rssiEntry = rssiEntryToString(r);
              if (rssiEntry != "") {
                timestamps.add(r.getTimeStamp());
                if (rssiEntry.equalsIgnoreCase("00:18:8E:22:24:57") || rssiEntry.
                    equalsIgnoreCase("00:18:8E:22:24:19") ) {
                  tags.add(rssiEntry);
                }
                else{
                  output.write(rssiEntryToString(r));
                }
              }
            }
            int tag_1 = Collections.frequency(tags, "00:18:8E:22:24:57");
            int tag_2 = Collections.frequency(tags, "00:18:8E:22:24:19");
```

```
                sumsWriter.write(location + "␣" + attenuation + "␣" +timestamps.size
                    ()+ "␣" + tag_1 + "␣" + tag_2  + "\n");
                timestamps.clear();
                tags.clear();
              }
            }
          }
        }
        sumsWriter.close();
        output.close();
      } catch ( IOException e ) {
        e.printStackTrace();
      }
    }
}
```

# B RSS analysis R-script

```
# This content is released under the (http://opensource.org/licenses/MIT)
# MIT License.

# Copyright (c) 2015 Ekahau, Ville Miekk-oja.

setwd("/home/miekkoja/dumptext_stuff")
options(stringsAsFactors=FALSE)
initial <- read.table("Antenni_sijainti_A.txt", header = TRUE)
ordered <- initial[order(initial$Place, initial$Attenuation, initial$TagMac,
    initial$RSS),]

#make matrix with average and variance
avg_var <- data.frame(Place = numeric(0), Attenuation = numeric(0), TagMac =
    character(0), AVG_RSS = numeric(0), VAR_RSS = numeric(0), Count = numeric(0))

for(i in seq_len(nrow(initial))) {
  if (i==1) {
    apu <-list()
    apu[[length(apu)+1]] <- ordered[i, "RSS"]
  }
  else {
      if (as.numeric(i) == as.numeric(nrow(initial))) {
      apu[[length(apu)+1]] <- ordered[i, "RSS"]

      #calculate average
      avg_sum = 0
      for(p in seq_along(apu)) {
        if (p!=0) {
          avg_sum = avg_sum + apu[[p]]
        }
      }
      pituus = length(apu)
      avg = avg_sum / pituus

      #calculate variance
      var_sum = 0
      for(e in seq_along(apu)) {
        if (e!=0) {
          var_sum = var_sum + (apu[[e]] - avg)^2
        }
      }
        var = var_sum / pituus
      newrow = c(ordered[(i-1), "Place"], ordered[(i-1), "Attenuation"], ordered[(i
          -1), "TagMac"], avg, var, pituus)
      avg_var[nrow(avg_var)+1,] <- newrow
      apu <-list()
      apu[[length(apu)+1]] <- ordered[i, "RSS"]

    }
    else if ((ordered[(i-1), "Place"] == ordered[i,"Place"]) & (ordered[(i-1), "
        Attenuation"] == ordered[i,"Attenuation"]) & (ordered[(i-1), "TagMac"] ==
        ordered[i,"TagMac"]) ) {
      apu[[length(apu)+1]] <- ordered[i, "RSS"]
    }
    else {
      #calculate average
      avg_sum = 0
      for(p in seq_along(apu)) {
        if (p!=0) {
          avg_sum = avg_sum + apu[[p]]
        }
      }
      pituus = length(apu)
      avg = avg_sum / pituus
      #calculate variance
```

```
      var_sum = 0
      for(e in seq_along(apu)) {
        if (e!=0) {
          var_sum = var_sum + (apu[[e]] - avg)^2
        }
      }
        var = var_sum / pituus

      newrow = c(ordered[(i-1), "Place"], ordered[(i-1), "Attenuation"], ordered[(i
          -1), "TagMac"], avg, var, pituus)
      avg_var[nrow(avg_var)+1,] <- newrow


      apu <-list()
      apu[[length(apu)+1]] <- ordered[i, "RSS"]
    }
  }
}

avg_var
plot(t(avg_var["Attenuation"]),t(avg_var["AVG_RSS"]))

#Calculate the average difference of the average RSS
#of the tags with zero attenuation:

avg_var_zero_attenuation <- avg_var[as.numeric(avg_var$Attenuation) == 0,]
sum_of_differences = 0.0
sum_of_counts = 0.0

for(i in seq(2, nrow(avg_var_zero_attenuation), 2)) {

  if (as.numeric(avg_var_zero_attenuation[(i), "Place"]) == 6) {
    next
  }
  current_avg = as.double(avg_var_zero_attenuation[(i), "AVG_RSS"])
  previous_avg = as.double(avg_var_zero_attenuation[(i-1), "AVG_RSS"])
  difference = abs(current_avg - previous_avg)
  current_amount = as.double(avg_var_zero_attenuation[(i), "Count"])
  previous_amount = as.double(avg_var_zero_attenuation[(i-1), "Count"])
  sum_of_differences = sum_of_differences + difference*(current_amount +
      previous_amount)
  sum_of_counts = sum_of_counts + current_amount + previous_amount
}
average_diff_of_avg_rss_between_tags = sum_of_differences / sum_of_counts
average_diff_of_avg_rss_between_tags
```

# C Full results of the RSS measurements

| Place | Attenuation | TagMac | AVG_RSS | VAR_RSS | Count |
|---|---|---|---|---|---|
| 1 | 0 | 00:18:8E:22:24:19 | -62.27 | 0.94 | 51 |
| 1 | 0 | 00:18:8E:22:24:57 | -74 | 0.98 | 41 |
| 1 | 5 | 00:18:8E:22:24:19 | -65.07 | 4.71 | 46 |
| 1 | 5 | 00:18:8E:22:24:57 | -76.04 | 1.29 | 48 |
| 1 | 10 | 00:18:8E:22:24:19 | -65.13 | 1.81 | 52 |
| 1 | 15 | 00:18:8E:22:24:19 | -70.7 | 0.78 | 46 |
| 1 | 20 | 00:18:8E:22:24:19 | -76.12 | 0.61 | 8 |
| 2 | 0 | 00:18:8E:22:24:19 | -65.26 | 12.71 | 50 |
| 2 | 0 | 00:18:8E:22:24:57 | -72.82 | 5.51 | 50 |
| 2 | 5 | 00:18:8E:22:24:19 | -65.96 | 5.7 | 47 |
| 2 | 5 | 00:18:8E:22:24:57 | -74.77 | 2.01 | 47 |
| 2 | 10 | 00:18:8E:22:24:19 | -70.76 | 3.94 | 49 |
| 2 | 15 | 00:18:8E:22:24:19 | -78 | 0 | 2 |
| 2 | 20 | 00:18:8E:22:24:57 | -77.3 | 1.31 | 20 |
| 3 | 0 | 00:18:8E:22:24:19 | -75.18 | 4.2 | 38 |
| 3 | 0 | 00:18:8E:22:24:57 | -76.03 | 3.32 | 37 |
| 3 | 5 | 00:18:8E:22:24:19 | -73.62 | 2.94 | 34 |
| 3 | 5 | 00:18:8E:22:24:57 | -77.26 | 0.82 | 35 |
| 3 | 10 | 00:18:8E:22:24:19 | -77.5 | 0.25 | 4 |
| 3 | 15 | 00:18:8E:22:24:19 | -79.33 | 0.89 | 3 |
| 3 | 15 | 00:18:8E:22:24:57 | -78 | 0 | 1 |
| 3 | 20 | 00:18:8E:22:24:19 | -80 | 0 | 1 |
| 4 | 0 | 00:18:8E:22:24:19 | -76.29 | 0.56 | 17 |
| 4 | 0 | 00:18:8E:22:24:57 | -75.44 | 0.63 | 52 |
| 4 | 5 | 00:18:8E:22:24:19 | -77 | 1.2 | 10 |
| 4 | 5 | 00:18:8E:22:24:57 | -79 | 4 | 2 |
| 4 | 10 | 00:18:8E:22:24:19 | -80.33 | 0.22 | 3 |
| 5 | 0 | 00:18:8E:22:24:19 | -74.29 | 1.92 | 7 |
| 6 | 0 | 00:18:8E:22:24:57 | -77.67 | 0.89 | 6 |
| 6 | 5 | 00:18:8E:22:24:57 | -81 | 0 | 1 |
| 7 | 0 | 00:18:8E:22:24:19 | -75.83 | 0.49 | 52 |
| 7 | 0 | 00:18:8E:22:24:57 | -75.58 | 1.09 | 57 |
| 7 | 5 | 00:18:8E:22:24:19 | -76.52 | 0.52 | 44 |
| 7 | 5 | 00:18:8E:22:24:57 | -77.25 | 1.19 | 20 |
| 7 | 10 | 00:18:8E:22:24:19 | -81 | 2 | 3 |
| 7 | 15 | 00:18:8E:22:24:19 | -80 | 0 | 2 |
| 8 | 0 | 00:18:8E:22:24:19 | -71.98 | 1.46 | 57 |
| 8 | 0 | 00:18:8E:22:24:57 | -75.38 | 1.48 | 8 |
| 8 | 5 | 00:18:8E:22:24:19 | -77.43 | 1.12 | 23 |
| 8 | 5 | 00:18:8E:22:24:57 | -77.4 | 1.04 | 5 |
| 8 | 10 | 00:18:8E:22:24:19 | -81.5 | 1.25 | 4 |
| 8 | 20 | 00:18:8E:22:24:57 | -81 | 0 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| 9 | 0 | 00:18:8E:22:24:19 | -67.52 | 2.49 | 58 |
| 9 | 0 | 00:18:8E:22:24:57 | -73.73 | 1.54 | 55 |
| 9 | 5 | 00:18:8E:22:24:19 | -72.25 | 1.21 | 59 |
| 9 | 5 | 00:18:8E:22:24:57 | -76.66 | 0.86 | 44 |
| 9 | 10 | 00:18:8E:22:24:19 | -77.5 | 1.25 | 4 |
| 9 | 10 | 00:18:8E:22:24:57 | -79 | 0 | 2 |
| 9 | 15 | 00:18:8E:22:24:19 | -79.8 | 3.76 | 5 |
| 10 | 0 | 00:18:8E:22:24:19 | -65.27 | 1.99 | 59 |
| 10 | 0 | 00:18:8E:22:24:57 | -72.04 | 0.63 | 57 |
| 10 | 5 | 00:18:8E:22:24:19 | -74.42 | 1.02 | 57 |
| 10 | 5 | 00:18:8E:22:24:57 | -78.75 | 0.44 | 8 |
| 10 | 10 | 00:18:8E:22:24:19 | -78 | 0 | 1 |
| 11 | 0 | 00:18:8E:22:24:19 | -71.39 | 1.11 | 57 |
| 11 | 0 | 00:18:8E:22:24:57 | -78 | 0 | 1 |
| 11 | 5 | 00:18:8E:22:24:19 | -76.19 | 0.91 | 53 |
| 11 | 5 | 00:18:8E:22:24:57 | -77.55 | 0.59 | 29 |
| 11 | 10 | 00:18:8E:22:24:19 | -78.75 | 1.69 | 4 |
| 11 | 15 | 00:18:8E:22:24:19 | -79 | 4 | 2 |
| 12 | 0 | 00:18:8E:22:24:19 | -71.35 | 1.63 | 57 |
| 12 | 0 | 00:18:8E:22:24:57 | -77.14 | 3.27 | 21 |
| 12 | 5 | 00:18:8E:22:24:19 | -72.5 | 2.57 | 56 |
| 12 | 5 | 00:18:8E:22:24:57 | -77.67 | 0.36 | 15 |
| 12 | 10 | 00:18:8E:22:24:19 | -75.53 | 0.89 | 47 |
| 12 | 15 | 00:18:8E:22:24:19 | -77.5 | 0.25 | 2 |
| 13 | 0 | 00:18:8E:22:24:19 | -71.71 | 5.17 | 58 |
| 13 | 0 | 00:18:8E:22:24:57 | -74.07 | 3.32 | 59 |
| 13 | 5 | 00:18:8E:22:24:19 | -67.7 | 0.57 | 56 |
| 13 | 5 | 00:18:8E:22:24:57 | -77.44 | 0.91 | 9 |
| 13 | 10 | 00:18:8E:22:24:19 | -74.11 | 0.63 | 56 |
| 13 | 15 | 00:18:8E:22:24:19 | -78 | 0 | 1 |
| 13 | 20 | 00:18:8E:22:24:19 | -81 | 1 | 2 |

# D BLE Scanner Android software

```
/**
 * This content is released under the (http://opensource.org/licenses/MIT)
 * MIT License.
 *
 * Copyright (c) 2015 Ekahau, Ville Miekk-oja.
 */

package com.example.btlescanner;
import java.io.BufferedWriter; import java.io.File;
import java.io.FileOutputStream; import java.io.FileWriter;
import java.util.ArrayList; import java.util.Date;
import android.net.Uri; import android.os.Bundle;
import android.os.Environment; import android.os.Handler;
import android.app.Activity; import android.app.ListActivity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothGatt;
import android.bluetooth.BluetoothGattCallback;
import android.bluetooth.BluetoothGattCharacteristic;
import android.bluetooth.BluetoothManager;
import android.bluetooth.BluetoothProfile; import android.content.Context;
import android.content.Intent; import android.content.pm.PackageManager;
import android.text.format.DateFormat; import android.text.format.Time;
import android.util.Log; import android.view.LayoutInflater;
import android.view.Menu; import android.view.MenuItem;
import android.view.View; import android.view.ViewGroup;
import android.widget.BaseAdapter; import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
  private BluetoothAdapter mBluetoothAdapter;
  private Handler scanHandler;
  public static final int MENU_EXITAPPLICATION = Menu.FIRST;
  private static final int REQUEST_ENABLE_BT = 1;
  private static final long SCAN_INTERVAL = 500;
  private static final long WAIT_BEFORE_START_SCAN = 0;
  TextView textStatus;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    scanHandler = new Handler();
    textStatus = (TextView) findViewById(R.id.textStatus);
    textStatus.setText("");

    //Check to determine whether BLE is supported on the device.
    if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_LE))
        {
      Toast.makeText(this, R.string.ble_not_supported,Toast.LENGTH_SHORT).show();
      finish();
    }

    //Initializes a Bluetooth adapter. For API level 18 and above,
    //get a reference to BluetoothAdapter through BluetoothManager.
    final BluetoothManager bluetoothManager = (BluetoothManager) getSystemService(
        Context.BLUETOOTH_SERVICE);
    mBluetoothAdapter = bluetoothManager.getAdapter();

    //Ensures Bluetooth is available on the device and it is enabled.
    //If not, displays a dialog requesting user permission to
    //enable Bluetooth.
    if (mBluetoothAdapter == null || !mBluetoothAdapter.isEnabled()) {
      Intent enableBtIntent = new Intent(
          BluetoothAdapter.ACTION_REQUEST_ENABLE);
```

```
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
  }

  scanHandler.postDelayed(startScan, WAIT_BEFORE_START_SCAN);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
  // Inflate the menu; this adds items to the action bar if it is present.
  getMenuInflater().inflate(R.menu.main, menu);
  menu.add(0, MENU_EXITAPPLICATION, 1, "Exit␣BTLE_Scanner␣App");
  return super.onCreateOptionsMenu(menu);
}

@Override
protected void onResume() {
  super.onResume();
  //Ensures Bluetooth is enabled on the device. If Bluetooth is not
  //currently enabled, fire an intent to display a dialog asking
  //the user to grant permission to enable it.
  if (!mBluetoothAdapter.isEnabled()) {
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
  }
}

private Runnable startScan = new Runnable() {

  @Override
  public void run() {
    mBluetoothAdapter.startLeScan(mLeScanCallback);
    // recursively scan to infinity
    scanHandler.postDelayed(startScan, SCAN_INTERVAL);
  }
};

private BluetoothAdapter.LeScanCallback mLeScanCallback =
  new BluetoothAdapter.LeScanCallback() {
    @Override
    public void onLeScan(final BluetoothDevice device, final int rssi, byte[]
        scanRecord) {
      runOnUiThread(new Runnable() {
        @Override
        public void run() {
          Time now = new Time();
          now.setToNow();
          String message = "" + device.getAddress();
          message = message + ":␣" + rssi + "␣" + now + "\n";
          textStatus = (TextView) findViewById(R.id.textStatus);
          textStatus.append(message);
          try {
            File myFile = new File(Environment.getExternalStorageDirectory().
                getPath() + "/BLE_ScanResult.txt");
            if (!myFile.exists())
              myFile.createNewFile();
            BufferedWriter bW;
            bW = new BufferedWriter(new FileWriter(myFile, true));
            bW.write(message);
            bW.newLine();
            bW.flush();
            bW.close();

          } catch (Exception e) {
            Toast.makeText(getBaseContext(), e.getMessage(),
            Toast.LENGTH_SHORT).show();
          }
        }
      });
```

```
        }
    };
}
```

# E BLEReadinginjector Java program

```
/**
 * This content is released under the (http://opensource.org/licenses/MIT)
 * MIT License.
 *
 * Copyright (c) 2015 Ekahau, Ville Miekk-oja, Johannes Verwijnen
 */

import java.io.DataInputStream; import java.io.DataOutputStream;
import java.io.File; import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.ArrayList; import java.util.HashMap;
import java.util.LinkedHashMap; import java.util.LinkedList;
import java.util.List; import java.util.Map; import java.util.Scanner;
import java.lang.Math.*;

public class BTReadingInjector {

  public static long SECONDS_FROM_EPOCH_TO_SURVEY = 1406386503;

  public static void main(String[] args) throws Exception {
    if (args.length == 0) {
      args = new String[3];
      args[0] = "BLE_ScanResult.txt";
      args[1] = "survey_WIFI.bin";
      args[2] = "survey_BT.bin";
    }
   if (args.length != 3) {
    System.out.println("Usage: BTReadingInjectorNew btmeasurements old_survey
        surveyfile");
    System.out.println("where btmeasurements is a BT measurement text file for
        input");
    System.out.println("and old_survey is an existing binary WIFI-survey file");
    System.out.println("and surveyfile is the name of the combined binary survey
        file (existing will be overwritten)");
    System.exit(-1);
   }
   File inFile = new File(args[0]);
   File wifiFile = new File(args[1]);
   File outFile = new File(args[2]);
   DataOutputStream dos = new DataOutputStream(new FileOutputStream(outFile));
   Scanner s = new Scanner(inFile);

   try {
     List<Long> scanTimes = new ArrayList<>();
     String BT_MAC = null;
     long start = 0;
     int scanIndex = -1;
     DataInputStream dis = new DataInputStream(new FileInputStream(wifiFile));
     Map<Long, String> BT_RSSIs = new LinkedHashMap<Long, String>();
     boolean firstTime = true;
     ArrayList<HashMap<Long, String>> tempList = new ArrayList<HashMap<Long, String
         >>();

     while (s.hasNextLine()) {
       String row = s.nextLine();
       if (!row.isEmpty()) {
         String[] fields = row.split(" ");
         BT_MAC = fields[0].substring(0, fields[0].length() - 1);
         String RSSI = fields[1];
         long timestamp = Long.parseLong(fields[2].substring(fields[2].lastIndexOf(
             ',') + 1, fields[2].length() - 1));
         if (firstTime == true) {
            HashMap<Long, String> currentValues = new HashMap<Long, String>();
            currentValues.put(timestamp, RSSI);
            tempList.add(currentValues);
```

```
        }
        else {
          boolean contains = false;

          //check if current timestamp is same as before:
          for(HashMap<Long, String> map: tempList) {
            if (map.containsKey(timestamp)) {
              contains = true;
            }
          }
          if (contains == false) {
            //new timestamp has been spotted -> calculate
            //average RSSI of the old ones, and add that to
            //BT_RSSIs, reset the tempList and add the newest
            //value there:
            int total = 0;
            int count = tempList.size();
            long timestamp1 = 0;

            for(HashMap<Long, String> map: tempList) {
                for (String rss: map.values()) {
                  int currentRss = Integer.parseInt(rss);
                  total = total + currentRss;
                }
                for (long time: map.keySet()) {
                  timestamp1 = time;
                }
            }
            float totalFloat = (float)total;
            float average = totalFloat / (count*1.0f);
            int average1 = Math.round(average);
            String average_string = Integer.toString(average1);

            //add average and their timestamp to BT_RSSIs:
            BT_RSSIs.put(timestamp1, average_string);
            tempList.clear();
            HashMap<Long, String> currentValues = new HashMap<Long, String>();
            currentValues.clear();
            currentValues.put(timestamp, RSSI);
            tempList.add(currentValues);
          }
          else {
            //current timestamp is already found in previous
            //timestamps -> add it to the tempList(with its RSS):
            HashMap<Long, String> currentValues = new HashMap<Long, String>();
            currentValues.put(timestamp, RSSI);
            tempList.add(currentValues);
          }
        }
        firstTime = false;
      }
    }

    //REMEMBER TO PUT THE LAST LINES TO BT_RSSIs:
    int total = 0;
    int count = tempList.size();
    long timestamp1 = 0;

    for(HashMap<Long, String> map: tempList) {
      for (String rss: map.values()) {
        int currentRss = Integer.parseInt(rss);
        total = total + currentRss;
      }

      for (long time: map.keySet()) {
        timestamp1 = time;
      }
    }
```

```
      float totalFloat = (float)total;
      float average = totalFloat / (count*1.0f);
      int average1 = Math.round(average);
      String average_string = Integer.toString(average1);
      //add average and their timestamp to BT_RSSIs:
      BT_RSSIs.put(timestamp1, average_string);
      List<Long> timestamps = new LinkedList(BT_RSSIs.keySet());
      int index = 0;
      long old_time = Long.MIN_VALUE;

      while (true) {
        long ts = dis.readLong();
        int scanIdx = dis.readInt();
        long measIdx = dis.readLong();
        int rssi = dis.readInt();
        int noise = dis.readInt();
        int channel = dis.readInt();

        if (old_time == Long.MIN_VALUE) {
          old_time = ts;
        } else if (ts != old_time) {
          long temp1 = Math.abs(timestamps.get(index) * 1000L - (
              SECONDS_FROM_EPOCH_TO_SURVEY*1000L + ts));
          long temp2 = Math.abs(timestamps.get(index + 1) * 1000L - (
              SECONDS_FROM_EPOCH_TO_SURVEY*1000L + ts));

          if (temp1 > temp2) {
            index++;
          }
          old_time = ts;
          dos.writeLong(ts);
          dos.writeInt(scanIdx);
          dos.writeLong(16);
          dos.writeInt(Integer.parseInt(BT_RSSIs.get(timestamps.get(index))));
          dos.writeInt(-2147483648);
          dos.writeInt(0);
          dos.flush();
        }
        dos.writeLong(ts);
        dos.writeInt(scanIdx);
        dos.writeLong(measIdx);
        dos.writeInt(rssi);
        dos.writeInt(noise);
        dos.writeInt(channel);
        dos.flush();
      }
    } catch (Exception e) {
      System.out.println(e.getMessage());

    } finally {
      dos.close();
      s.close();
    }
  }
}
```